Enabling Cross-Technology Coexistence for ZigBee Devices Through Payload Encoding

Junmei Yao, Haolang Huang, Jiongkun Su, Ruitao Xie, Xiaolong Zheng, and Kaishun Wu

***Abstract*—With the rapid growth of Internet of Things, the number of heterogeneous wireless devices working in the same frequency band increases dramatically, leading to severe cross-technology interference. To enable coexistence, researchers have proposed a large number of mechanisms to manage interference. However, existing mechanisms have severe modifications in either the physical or MAC (medium access control) layers, making them very different from the standard. In this paper, we design and implement SledZig to boost cross-technology coexistence for low-power devices through both enabling more transmission opportunities and avoiding interference. SledZig is fully compatible with the standard in both physical and MAC layers. It decreases the WiFi signal power on the channel of low-power devices while keeps the WiFi transmission power unchanged, through making constellation points on the overlapped subcarriers have the lowest power, which can be achieved by just encoding the WiFi payload. We implement SledZig on hardware testbed and evaluate its performance under different settings. Experiment results show that SledZig can effectively increase ZigBee transmissions and improve its performance over a WiFi channel under various WiFi data traffic, with as low as 6.94% WiFi throughput loss.**

***Index Terms*—Heterogeneous wireless networks, coexistence, WiFi, ZigBee.**

I. Introduction

The prosperity of Internet of Things (IoT) increases the number of wireless devices exponentially. Wireless devices adopt heterogeneous wireless technologies, as each technology has its own suitable application scenarios due to its strengths and weaknesses. In the crowded ISM (industrial, scientific and medical) frequency band, the heterogeneous wireless devices inevitably work on the overlapped channels, leading to severe cross-technology coexistence problem.

WiFi and ZigBee are the two most common wireless technologies in IoT. WiFi is used for wireless local area networks (WLAN), while its market has stable increase now and in the future. Cisco predicts that the number of WiFi hotspots will reach 628 Million by 2023 [1]. Meanwhile, ZigBee plays an important role in providing low cost, low data rate, and low energy consumption characteristics for wireless sensor networks. The ZigBee market also increases steadily these years. It was valued at USD 2.81 Billion in 2018 and is projected to reach USD 5.38 Billion by 2026 [2]. WiFi and ZigBee has asymmetry power levels. The ZigBee signal is always transmitted at less than $1mW$ for energy saving, while the WiFi signal is transmitted at up to 100 mW for large coverage. Meanwhile, when the devices are contending channel, WiFi has higher priority than ZigBee and can always win the channel for data transmission, due to their MAC layer design. Thus, the WiFi devices induce severe coexistence problems to ZigBee devices, through either prohibiting the ZigBee devices from data transmission or interfering the ongoing ZigBee data transmission.

The coexistence problem has attracted much research interest in past years. The related works can be categorized into two groups: cross-technology interference avoidance and interference resistance. Interference avoidance mechanisms always mitigate cross-technology interference (CTI) through designing physical (PHY) or MAC layer protocols. For example, EmBee [3] lets a WiFi device identify the channel of ZigBee signals and then reserves the corresponding channel for ZigBee transmission through designing null subcarriers. Interference resistance mechanisms try to recover the collided signal through PHY layer design, such as CrossZig [4], which utilizes packet merging and adaptive forward error correction (FEC) coding to recover packets under CTI. Both kinds of mechanisms require modifications on either the MAC layer or the PHY layer, thus are hard to be deployed to real networks.

In this paper, we propose SledZig, a subcarrier-level energy decreasing mechanism on WiFi to boost ZigBee transmission. SledZig is fully compatible with the standard PHY and MAC layer processes of both WiFi and ZigBee. It decreases the WiFi signal energy on the ZigBee channel while keeps the WiFi transmission power unchanged, through exploiting the features of QAM (quadrature amplitude modulation) modulation in WiFi. QAM is a combination of phase and amplitude modulation methods, making the QAM constellation points have different power levels. Through payload encoding which inserts extra bits

Manuscript received 30 June 2023; revised 1 November 2023; accepted 18 December 2023. Date of publication 21 December 2023; date of current version 2 July 2024. This work was supported in part by the China NSFC under Grants 62072317, 62272316, 62072050, and U2001207, in part by the Guangdong NSF under Grant 2023A1515010663, in part by the Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things under Grant 2023B1212010007, and in part by the project of DEGP under Grant 2023KCXTD042. Recommended for acceptance by V.P. Ranga Rao. *(Corresponding author: Ruitao Xie.)*

Junmei Yao, Haolang Huang, Jiongkun Su, and Ruitao Xie are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: ms.yaojunmei@gmail.com; huanghaolang2020@email.szu.edu.cn; 2100271095@email.szu.edu.cn; xie@szu.edu.cn).

Xiaolong Zheng is with the College of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: zhengxiaolong@bupt.edu.cn).

Kaishun Wu is with the Hong Kong University of Science and Technology (Guangzhou), Guangzhou 510275, China, and also with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: wuks@hkust-gz.edu.cn).

Digital Object Identifier 10.1109/TMC.2023.3345830

to original WiFi data bits, the QAM points on subcarriers overlapped with the ZigBee channel have the lowest power, while those out of the ZigBee channel remain unchanged, leading to up to $15dB$ energy decreasing on the ZigBee channel. With this energy decreasing, the ZigBee network performance can be improved dramatically through both enabling more transmission opportunities and avoiding interference. Actually, the idea of using payload encoding to change the transmitting signal starts from WEBee [5], which makes a WiFi device transmit an emulated ZigBee signal through payload encoding, thus to achieve WiFi to ZigBee cross-technology communication (CTC). After that, we see some other papers use the similar idea to achieve CTC in other contexts [6], [7], [8], [9]. This paper is the first to use payload encoding to improve the coexistence network performance.

In order to achieve SledZig, there are several important issues that need to be addressed. At first, the transmitter needs to determine the values and positions of extra bits in advance, so as to perform correct payload encoding; we follow the reverse WiFi transmission process step by step to achieve this goal (Section VI). At second, the ZigBee channel needs to be obtained before payload encoding as it affects the positions and values of extra bits; thus, we propose ZigBee channel identification which utilizes the frequency spectrum features to identify ZigBee signals and the corresponding channels (Section V). Finally, as the payload encoding can only reduce the signal power of payload in a WiFi packet, the preamble at the head of the packet is still with high power, causing burst interference to the ZigBee transmission; we design Reed-Solomon (RS) code for ZigBee to combat this interference, thus further improve ZigBee performance (Section VII).

From the perspective of usage, SledZig is quite simple. The WiFi transmitter first conducts payload encoding according to the identified ZigBee channel and QAM modulation, so as to generate the transmit bits. When the transmit bits are passed through the standard WiFi transmission process, the signal energy on the ZigBee channel can be automatically decreased, thus to boost ZigBee transmissions. The WiFi receiver can easily obtain the original data bits through removing the extra bits from the received bits. Meanwhile, the ZigBee devices can adopt RS encoding and decoding for the ZigBee payload to combat interference from WiFi preamble, so as to further improve the performance. This process is fully compatible with the current standard. Considering that all devices should comply with the standard for data transmission in real networks, SledZig has the big advantage compared with previous works which need to make either physical or MAC layer modifications.

This paper makes the following main contributions:

- We design SledZig, a subcarrier-level energy decreasing mechanism on WiFi to decrease the signal power on ZigBee channels through payload encoding, thus to increase the ZigBee network performance from both enabling more transmission opportunities and avoiding CTI. To the best of our knowledge, SledZig is the first mechanism that improve the ZigBee performance through just encoding the WiFi payload. It is compatible with the standard in both PHY and MAC layers, and has the potential to be deployed to real networks.
- To achieve SledZig, we address some unique issues. We propose a general payload encoding process through following the reverse WiFi transmission process step by step. We also make WiFi devices identify ZigBee channels correctly to perform payload encoding. We further design RS code for ZigBee to combat the burst interference from WiFi preamble, thus to improve the ZigBee performance.
- We implement SledZig on hardware testbed based on USRP N210 and TelosB platforms. Experimental results indicate that SledZig can decrease the WiFi signal power on a ZigBee channel by up to $15dB$. Meanwhile, it can improve the ZigBee performance dramatically with as low as 6.94% WiFi throughput loss.

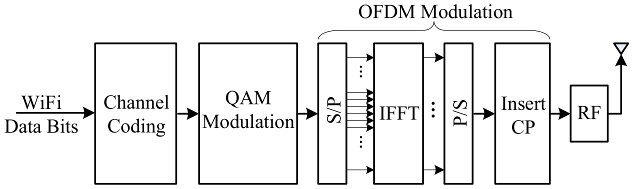


Fig. 1. Standard WiFi transmission process.

This paper is organized as follows: Section II briefly describes the basic knowledge of WiFi transmission and the main differences between WiFi and ZigBee in the PHY and MAC layers. Section III illustrates the coexistence problem and presents the opportunity to solve the problem. Section IV gives the system overview. Section V presents the design of ZigBee channel identification. Section VI presents the detailed design of payload encoding. Section VII proposes a mechanism to combat the WiFi preamble interference. Section VIII evaluates the performance of SledZig comparing with the standards through hardware experiments. Section IX introduces related works. Section X concludes this paper.

II. Background

In this part, we introduce the background knowledge that is important for the SledZig design.

A. WiFi Transmission

Fig. 1 depicts the standard WiFi transmission process. The data bits are first passed through the channel coding module, and transformed to complex symbols after QAM modulation; the QAM constellation points are then mapped onto OFDM (orthogonal frequency division multiplexing) subcarriers after the S/P (serial-to-parallel) module, and output as the time-domain OFDM symbols after IFFT (inverse fast fourier transform) and P/S (parallel-to-serial) processes; each OFDM symbol is inserted with CP (cyclic prefix) to eliminate the inter-symbol interference; the signal will finally be transmitted after RF front end.

It is worth noting that OFDM makes a device transmit signals on multiple orthogonal subcarriers which are closely spaced to carry data in parallel. In the WiFi system, each $20MHz$ WiFi channel is divided into 64 subcarriers, including 48 data subcarriers, 4 pilot subcarriers and 12 null subcarriers, as shown in Fig. 2. According to the standard process, each data subcarrier

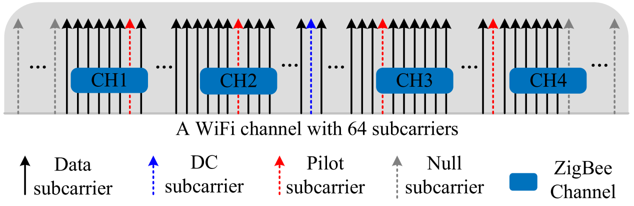


Fig. 2. Illustration of the WiFi channel overlapping with four ZigBee channels.

carries a QAM point, which is generated from the original data bits after channel coding, QAM modulation and S/P module. Thus, we cannot make some data subcarriers be null directly to avoid interference to ZigBee.

B. Differences of WiFi and ZigBee

1) The PHY Layer Specifications: WiFi and ZigBee working in the 2.4 GHz ISM band have distinct specifications. They adopt different PHY layer technologies, as WiFi adopts OFDM and QAM modulations but ZigBee adopts DSSS (direct sequence spread spectrum) and OQPSK (offset quadrature phase shift keying) modulations. Besides that, they have different channel bandwidth. ZigBee has sixteen 2 MHz channels with 5 MHz channel spacing, numbering from 11 to 26. WiFi has thirteen 20 MHz channels with 25 MHz channel spacing.[1] Thus, one WiFi channel overlaps with four ZigBee channels. Each WiFi channel which contains 64 subcarriers overlaps with four ZigBee channels in the same pattern, as shown in Fig. 2. For the ease of description in the following part, we call the four ZigBee channels as CH1, CH2, CH3 and CH4 for short. We see that CH1-CH3 overlap with a pilot subcarrier and CH4 overlaps with null subcarriers.

Moreover, the two kinds of devices have asymmetry transmission power. ZigBee devices have the transmission power of no more than $0dBm$ to cut down energy consumption, while the WiFi transmission power can be up to 20 dBm with the purpose of large coverage.

We note that WiFi can work on both 2.4 GHz and 5 GHz frequency bands. Actually, IEEE 802.11ac (called WiFi 5) [10] only supports the 5 GHz band, but IEEE 802.11ax (called WiFi 6) [11] still supports dual frequency bands due to the larger coverage of 2.4 GHz band. Thus, some traffic is diffused on the 5 GHz band. However, we believe that the coexistence problem is still important on the 2.4 GHz band with the increase of data traffic and the number of IoT devices [3], [4], [12], [13], [14], [15], [16], [17].

2) The MAC Layer Specifications: Both the WiFi and ZigBee networks adopt the CSMA/CA mechanism to contend the channel. The detailed CSMA/CA mechanism is shown in Fig. 3. When a device begins to transmit a data packet, it first waits for DIFS time; if the channel is idle during DIFS, the device then waits for a random duration which consists of multiple backoff timeslots to contend for the channel; the backoff timer is decreased by one when the channel is idle for a backoff slot, and is frozen when the channel is busy; the device can finally transmit a data packet if the backoff timer reaches zero. During DIFS or each backoff timeslot, the device should perform CCA (clear channel assessment) to determine whether the channel is idle. The channel is determined to be idle if the detected signal energy is below a predefined threshold; otherwise it is busy.

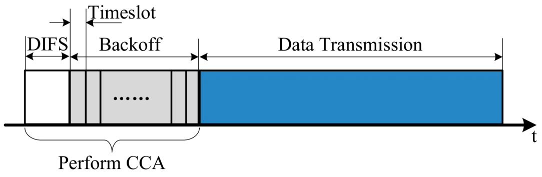
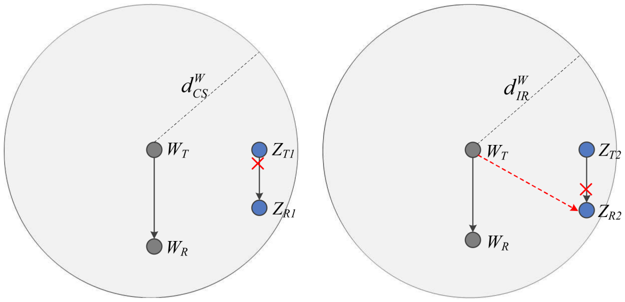


Fig. 3. Illustration of CSMA/CA.



(a) ZigBee devices within the WiFi carrier sense range d^W_{CS} are prohibited to transmit data.

(b) The ZigBee transmission is interfered by the WiFi transmission.

Fig. 4. Two scenarios that WiFi affects the ZigBee performance. Here the WiFi carrier sense range d^W_{CS} indicates the range within which a ZigBee device determines the channel to be busy due to the WiFi transmission, the WiFi interference range d^W_{IR} indicates the range within which a ZigBee link will be interfered by the WiFi transmission.

The main difference here between WiFi and ZigBee is that, the WiFi DIFS is $28\mu s$ [18] while ZigBee DIFS is $320\mu s$ [19], meanwhile, WiFi backoff slot is 9 or 20 μs while ZigBee backoff slot is $320\mu s$. This leads to extreme unfairness in the channel competition, as the WiFi device can always win the channel for transmission.

III. Motivation

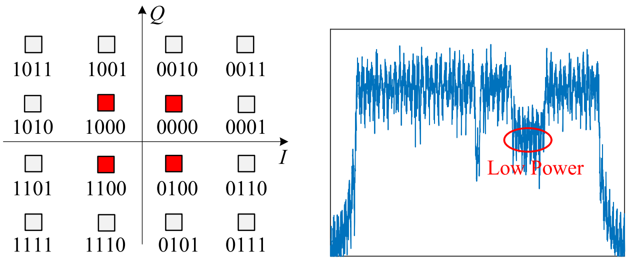
Here we first illustrate the cross-technology coexistence problem, then explain the opportunity on SledZig design.

A. Cross-Technology Coexistence Problem

The WiFi and ZigBee differences on PHY and MAC layers lead to severe cross-technology coexistence problem. Actually, with the asymmetry transmission power and MAC parameters, WiFi always affects the ZigBee network performance from two scenarios.

The first scenario lies in the fact that the high WiFi transmission power leads to a large carrier sense range d^W_{CS} and prohibits some ZigBee transmissions. As shown in Fig. 4(a), when the WiFi link $W_T \longrightarrow W_R$ and ZigBee link $Z_{T1} \longrightarrow Z_{R1}$ coexist in the network, the ZigBee device Z_{T1} is always prohibited from transmitting data to Z_{R1}. The reason comes from the

[1] The WiFi channel can be up to 40 MHz in 802.11n and 160 MHz in 802.11ax. This paper focuses on the 20 MHz channel, while the similar idea can be easily extended to wider channel scenarios.



(a) QAM-16 constellation points (b) WiFi Frequency spectrum

Fig. 5. Example of the QAM-16 lowest points and the frequency spectrum when all the overlapped subcarriers are filled with the lowest points.

unfairness in channel competition. As discussed in the previous part, the duration of WiFi DIFS or backoff timeslot is much shorter than that of ZigBee. Thus, when both W_T and Z_{T1} have data packets for transmission and contend the channel, W_T can always win, making ZigBee with extremely poor performance in this situation. Our preliminary experiments indicate that, the ZigBee link can proceed its data transmission only when the WiFi link is very unsaturated, that is, the WiFi application layer data rate should be below 20% of the PHY layer data rate.

The second scenario is that the WiFi transmission may interfere with the ZigBee transmission. As shown in Fig. 4(b), when the ZigBee link $Z_{T2} \longrightarrow Z_{R2}$ proceeds its data transmission, it still has a high probability to be interfered by the WiFi transmission $W_T \longrightarrow W_R$, since it is within the WiFi interference range d_{IR}^W. Here Z_{T2} may transmit its data packets either because it is out of d_{CS}^W of the WiFi link or because it wins the channel although it is within d_{CS}^W. The strong WiFi signal can easily interfere with the ZigBee transmission.

B. Opportunity

Our analysis on the two scenarios in Fig. 4 reveals that, decreasing the WiFi transmission power will obviously increase the ZigBee network performance. In Fig. 4(a), the WiFi carrier sense range d_{CS}^W will be shortened, allowing the ZigBee device Z_{T1} to be out of d_{CS}^W and have the opportunity to transmit data to Z_{R1}. In Fig. 4(b), the signal from W_T with lower power will have less interference on the ZigBee link $Z_{T2} \longrightarrow Z_{R2}$, leading to successful ZigBee transmissions.

One intuitive way to decrease the WiFi signal power is to adjust the transmit gain to decrease the transmission power, but it will obviously decrease the WiFi performance significantly. Some other methods try to reserve the channel for ZigBee, such as EmBee [3] which designs null subcarriers on the overlapped channel; however, these methods are standard-incompatible as they require PHY layer modifications.

We observe that the WiFi power on the overlapped subcarriers can be decreased through designing low-power constellation points. As shown in Fig. 1, a WiFi device conducts QAM modulation before the OFDM module. QAM modulation is a combination of phase and amplitude modulations. Fig. 5(a) shows the QAM-16 constellation points, each of which represents four data bits. Among the 16 points, the red points have the lowest power. When the QAM points on the overlapped subcarriers are all the red ones, the signal power in the ZigBee channel can be reduced significantly, as shown in Fig. 5(b). Since this method keeps the WiFi transmission power unchanged, it has limited impact on the WiFi performance.

How much power can be decreased through this way can be derived theoretically. Specifically, the QAM-M modulation encodes groups of $log_2 M$ bits into M constellation points. Each point is a complex symbol which can be denoted as $\mathbf{s}_i = (I_i, Q_i)$, where $I_i, Q_i \in \{\pm(2 \times m - 1)\}$, $i \in [1, M]$ and $m \in [1, \frac{log_2 M}{2}]$. In each QAM modulation, the four lowest points are always $(\pm 1, \pm 1j)$. That means, the low power $P_{low} = 2$. Considering that each point has the equal probability to show in a packet, the average power level of the WiFi signal is $P_{avg} = \sum_i \mathbf{s}_i^2/M$. Thus, the power decreased through putting lowest points on the overlapped subcarriers is calculated as P_{avg}/P_{low}. More concretely, that value under QAM-16, QAM-64 and QAM-256 is 7.0dB, 13.2 dB and 19.3 dB respectively.

We note that this idea is totally different from previous works which use lower-level QAM modulation to combat interference in WiFi networks, such as the rate adaptation mechanisms [20], as lower-level QAM modulation requires lower SNR (Signal to Noise Ratio) to demodulate packets. These works do not decrease the WiFi signal power and cannot reduce the WiFi interference to ZigBee.

IV. System Overview

The goal of SledZig is to encode the WiFi payload through inserting extra bits to the WiFi data bits, so as to generate the transmit bits; when the transmit bits are passed through the standard WiFi transmission process, the overlapped OFDM subcarriers are filled with the lowest constellation points to decrease the signal power in the ZigBee channel. Fig. 6 shows an overview of SledZig. The white blocks represent the standard WiFi transmission process, while the grey blocks are added for SledZig design. The blue dashed line indicates the process of determining extra bits in the design, and a general payload encoding algorithm will be summarized according to this process.

There are two key issues that need to be addressed to achieve this goal. The first is ZigBee channel identification, with which the WiFi transmitter can determine the overlapped subcarriers and fill them with the lowest QAM points. The second is payload encoding which determines where and what extra bits should be inserted into the WiFi data bits, according to the ZigBee channel and QAM modulation type. In the following parts, we will illustrate ZigBee channel identification in Section V, and illustrate payload encoding in Section VI.

V. ZigBee Channel Identification

ZigBee channel identification is the first step in SledZig design. With this information, a WiFi transmitter can determine which subcarriers need to carry the lowest QAM points. We exploit the frequency spectrum features of ZigBee signals in this process. As shown in Fig. 7(a), ZigBee signals in different channels possess distinguishable frequency spectrum features. The signal power within each ZigBee channel is significantly stronger than the surrounding signal power. In addition, the WiFi

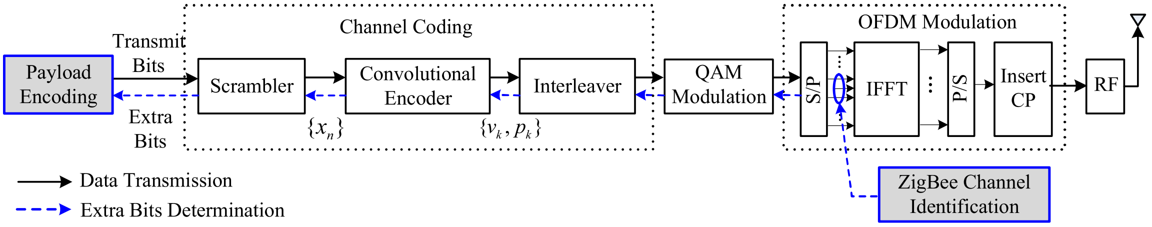
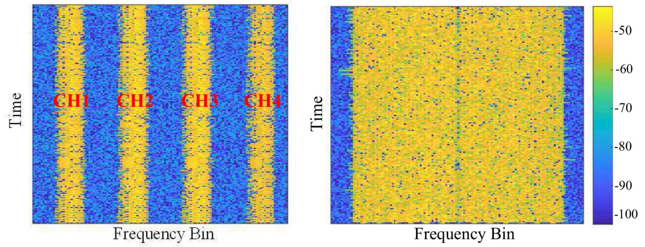


Fig. 6. Overview of SledZig.



(a) ZigBee

(b) WiFi

Fig. 7. Spectrogram of ZigBee and WiFi signals in a $20MHz$ channel.

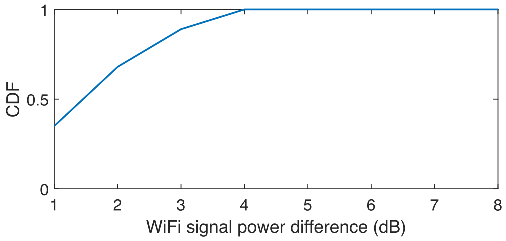


Fig. 8. WiFi signal power difference in each $1MHz$ band.

signals have very different spectrum features due to wider bandwidth, as shown in Fig. 7(b); thus, it is feasible to differentiate the two kinds of signals through these features.

To implement ZigBee channel identification, we let a WiFi device collect a sequence of N samples for each packet. It calculates the frequency spectrum of this sequence through conducting FFT (Fast Fourier Transform), and the signal power for each frequency bin is denoted by $\{P_k\}$, where $k = 1, \ldots, N$. It then identifies ZigBee and the corresponding channel through analyzing $\{P_k\}$. In order to consider the coexistence of other heterogeneous signals, such as Bluetooth with $1MHz$ channels which may overlap with ZigBee channels, we further divide each $2MHz$ ZigBee channel into four $0.5MHz$ subchannels. Then the averaged signal power within the ith subchannel of the jth ZigBee channel can be calculated as $P_Z^{CH_j^i} = \sum_{k=L^{CH_j^i}}^{R^{CH_j^i}} P_k$, where $L^{CH_j^i}$ and $R^{CH_j^i}$ indicate the left and right edge of the ith subchannel in the jth ZigBee channel, j from 1 to 4 indicate ZigBee channels from CH1 to CH4. Similarly, the adjacent $1\,MHz$ frequency band in the left and right side of the jth ZigBee channel can be calculated in the same way, which is denoted as $P_L^{CH_j}$ and $P_R^{CH_j}$. Only if all the inequalities $\frac{P_Z^{CH_j^i}}{P_L^{CH_j^i}} > \beta_Z$ and $\frac{P_Z^{CH_j^i}}{P_R^{CH_j^i}} > \beta_Z$ $(i \in [1, 4])$ hold, we can determine that there is a ZigBee signal in the jth channel. Here β_Z is the threshold to determine whether the signal power within a ZigBee channel is higher than the adjacent $1\,MHz$ band.

In this work, we intend to minimize the probability that a WiFi signal is mistakenly identified as ZigBee, as this will lead to unnecessary payload encoding at the WiFi transmitter. This goal can be achieved through setting β_Z. According to the aforementioned analysis, when a sequence of WiFi samples is used to calculate $\frac{P_Z^{CH_j^i}}{P_L^{CH_j^i}} > \beta_Z$ and $\frac{P_Z^{CH_j^i}}{P_R^{CH_j^i}} > \beta_Z$, none of the inequalities could hold for each combination of i and j, so that the device can determine that this is not a ZigBee signal. However, the result is highly related to the value of β_Z. For a normal WiFi signal, the QAM points on each $312.5KHz$ subcarrier are random, leading to significant differences in the averaged signal power in each $0.5\,MHz$ frequency bands. We collect 100 WiFi packets and calculate the signal power differences, and the corresponding cumulative distributed function (CDF) is shown in Fig. 8. We see that the difference is below $4dB$ with the probability of 100%. Thus, we set β_Z as $4\,dB$ in the identification process. We note that this threshold can also work for a WiFi signal with SledZig design, where the WiFi signal power within the ZigBee channel is much lower than the adjacent $1\,MHz$ frequency band; this WiFi signal also cannot be identified as a ZigBee signal as the inequalities do not hold.

VI. Payload Encoding

This section illustrates the detailed design of payload encoding, which can be achieved through following the reverse WiFi transmission process step by step to determine the extra bits, as shown in Fig. 6.

A. QAM Points

According to the design, the QAM points on the overlapped subcarriers should be the four ones with lowest power. For QAM-16, each point carries four bits, and only two bits are significant to make the power lowest. We call them as significant bits, as the shadowed ones shown in Table I. Similarly, each QAM-64 and QAM-256 point has four and six significant bits, respectively. The extra bits should be inserted only to make the significant bits be the designated ones, while the other bits in the QAM points can be arbitrary ones.

TABLE I
ILLUSTRATION OF SIGNIFICANT BITS

	QAM-16	QAM-64	QAM-256
Bits	0 0 0 0	0 1 0 0 1 0	0 1 0 0 0 1 0 0
	0 1 0 0	0 1 0 1 1 0	0 1 0 0 1 1 0 0
	1 0 0 0	1 1 0 0 1 0	1 1 0 0 0 1 0 0
	1 1 0 0	1 1 0 1 1 0	1 1 0 0 1 1 0 0

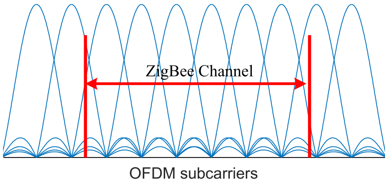


Fig. 9. Illustration of OFDM subcarriers overlapping with a ZigBee channel.

B. Overlapped Subcarriers

The more subcarriers used, the greater the impact on WiFi performance, since more extra bits should be inserted into the original WiFi data bits. Here the question is how many subcarriers are required for each ZigBee channel to achieve the lowest power.

The ZigBee channel is $2MHz$, while each OFDM subcarrier occupies $312.5KHz$. It is easy to take for granted that the number of overlapped subcarriers is $\lceil \frac{2MHz}{312.5KHz} \rceil = 7$. However, this will lead to suboptimal performance. As shown in Fig. 9, the OFDM signal contains multiple closely spaced orthogonal subcarriers. Each subcarrier still has energy leaked into the adjacent subcarriers. Thus, besides the six subcarriers fully overlapped with a ZigBee channel, the two adjacent subcarriers should also be filled with the lowest points. Therefore, we let each ZigBee channel overlap with eight subcarriers, among which one is pilot subcarrier in CH1-CH3, and three are null subcarriers in CH4.

C. Scrambler and Interleaver

The channel coding process includes interleaver, convolutional encoder and scrambler. Interleaver is used in wireless communication system to reduce the decoding errors, and SledZig design here is to generate the significant bits before interleaver through deinterleaving, according to those bits before QAM modulation. As shown in Fig. 6, we denote the significant bits before interleaver as $\{v_k, p_k\}(k \in [1, K])$, where v_k and p_k indicate the value and position of the kth significant bit. It is worth mentioning that, this process brings additional bonus for SledZig: the significant bits which are gathered together before deinterleaving are scattered to different locations far away, providing feasibility for the extra bits determination in convolutional encoding.

Scrambler is used to avoid long sequences of bits with the same value. SledZig design for this module is to obtain the transmit bits according to the scrambled transmit bits $\{x_n\}$. Since both modules are one-by-one mapping from input bits to output bits, the reverse processes for SledZig are quite easy.

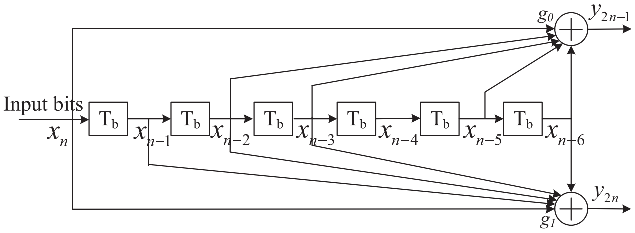


Fig. 10. Process of 1/2-rate convolutional encoding.

TABLE II
EXAMPLE OF SIGNIFICANT BITS IN THE FIRST OFDM SYMBOL

k	1	2	3	4	5	6	7
p_k	29	30	41	42	77	78	89
n	15	15	21	21	39	39	45
k	8	9	10	11	12	13	14
p_k	90	125	138	172	173	183	186
n	45	63	69	86	87	92	93

D. Convolutional Encoder

With QAM modulation and the ZigBee channel which determines the overlapped subcarriers, the significant bits $\{v_k, p_k\}$ after convolutional encoder are known by the WiFi transmitter. The main objective of SledZig design here is to determine the values and positions of extra bits before convolutional encoder, according to the original data bits and significant bits, as shown in Fig. 6. This process is challenging because convolutional encoder adds redundancy to the data bits, and it cannot generate arbitrary bit sequence. We achieve this goal through analyzing the convolutional encoding process and summarizing its characteristic to determine the extra bits.

The 802.11 standard recommends several coding rates under each QAM modulation, leading to different WiFi data rates. The 1/2-rate encoding is the basic process in convolutional encoding, where one input bit generates two output bits. The other coding rates like 2/3, 3/4 and 5/6 are achieved by employing puncturing on the 1/2-rate coded bits: some of the coded bits are omitted to increase the coding rate. Here we focus on the 1/2-rate encoding, the process for other coding rates are similar.

The 1/2-rate convolutional encoding process is shown in Fig. 10. It uses two generator polynomials $g_0 = (1011011)_2$ and $g_1 = (1111001)_2$. One input bit x_n triggers two coded bits y_{2n-1} and y_{2n}. The output coded bits are determined by not only the present input bit x_n but also a small number of previous bits from x_{n-1} to x_{n-6}. For the easy of description, we let $X_n = [x_n\, x_{n-1}\, x_{n-2}\, x_{n-3}\, x_{n-4}\, x_{n-5}\, x_{n-6}]'$. Then this one step encoding process to generate two output bits can be formulated as

$$\begin{aligned} g_0 \times_{GF(2)} X_n &= y_{2n-1}, \\ g_1 \times_{GF(2)} X_n &= y_{2n}, \end{aligned} \tag{1}$$

where $GF(2)$ means the calculation is in the Galois Field GF(2).

We have the significant bits $\{v_k, p_k\}$ after encoder, then the extra bits in the uncoded bits $\{x_n\}$ can be determined through (1) one by one. To make the description easier, we list an example of the significant bits in the first OFDM symbol in Table II,

Algorithm 1: Transmit Bits Generation Process.

```
Input  : Data bits $\{x'_i\}, i \in [1, N']$;
         Significant bits $\{v_k, p_k\}, k \in [1, K]$.
Output: Transmit Bits $\{x_n\}, n \in [1, N]$.
$k \leftarrow 1$;  $n \leftarrow 1$;  $etr_0 \leftarrow 0$;  $etr_1 \leftarrow 0$; $tmp \leftarrow 0$.
while $i \leq N'$ do
    if $(2n+1)$==$p_k$ and $2n+2$==$p_{k+1}$ then
        $X_{n+1} = [etr_0\ etr_1\ x_{n-1}\ x_{n-2}\ x_{n-3}\ x_{n-4}\ x_{n-5}]'$;
        $y_{2n+1} \leftarrow v_k$, $y_{2n+2} \leftarrow v_{k+1}$;
        Calculate $etr_0$ and $etr_1$ through Eq. 1;
        $x_n \leftarrow etr_1$;
        $n \leftarrow n+1$;
        $x_n \leftarrow etr_0$;
        $n \leftarrow n+1$; $k \leftarrow k+2$;
    else if $(2n-1)$==$p_k$ or $2n$==$p_k$ then
        $X_n = [etr_0\ x_{n-1}\ x_{n-2}\ x_{n-3}\ x_{n-4}\ x_{n-5}\ x_{n-6}]'$;
        if $(2n-1)$==$p_k$ then
            $y_{2n-1} \leftarrow v_k$.
        else
            $y_{2n} \leftarrow v_k$.
        Calculate $etr_0$ through Eq. 1;
        $x_n \leftarrow etr_0$;
        $n \leftarrow n+1$; $k \leftarrow k+1$.
    else
        $x_n \leftarrow x'_i$;
        $i \leftarrow i+1$;
        $n \leftarrow n+1$.
```

TABLE III
WHETHER SIGNIFICANTS BITS CAN BE SATISFIED WHEN CONSIDERING TWO ZIGBEE CHANNELS

Parallel Channels	QAM-16		QAM-64			QAM-256	
	1/2	3/4	2/3	3/4	5/6	3/4	5/6
CH1&CH2	✓	✓	✓	✓	✓	✓	✓
CH1&CH3	✓	✓	✓	✓	✓	✓	✓
CH1&CH4	×	×	✓	✓	✓	✓	✓
CH2&CH3	✓	✓	✓	✓	✓	×	×
CH2&CH4	✓	✓	✓	✓	✓	✓	✓
CH3&CH4	×	✓	✓	✓	✓	×	✓

where QAM-16 is adopted and the ZigBee channel is CH2. In this situation, there are $4 \times 48 = 192$ bits in each OFDM symbol after encoder, corresponding to 96 bits before encoder. There are 14 significant bits in each OFDM symbol, while their positions are listed as p_k in Table II. The significant bits have two situations, which are very important for the following analysis. One situation is that, given a n, either y_{2n-1} or y_{2n} in (1) is a significant bit, and the other one can be arbitrary bit, such as the case of $k = 9$, where $n = 63$ and $p_k = 2n - 1 = 125$ in Table II. We call this kind of bit as *single significant bit*. The other situation is that, both the two bits y_{2n-1} and y_{2n} are significant bits, such as the case of $k = 1$ and $k = 2$, where $n = 15$. We call this kind of bits as *twin significant bits*.

For the case of *single significant bit*, we let x_n be the extra bit, which should be inserted to make (1) hold. Here the bits from x_{n-6} to x_{n-1} may be scrambled WiFi data bits or extra bits determined in the previous steps, they are all known in the current step. x_n can be obtained easily through solving the corresponding equation in (1). Thus, the *single significant bit* can be satisfied through inserting one extra bit to the WiFi data bits in the designated positions.

For the case of *twin significant bits*, two extra bits are required to be unknowns in X_n to make (1) hold. We let x_n and x_{n-1} be the extra bits, and they can be determined through solving (1). We note that the bit x_{n-1} are also used to calculate the previous coded bits from $y_{2(n-1)-1}$ to $y_{2(n-1)}$. Once there are *single significant bit* or *twin significant bits* among them, (1) may have no solution, as there will be three or four equations together but only two unknowns. This abnormal situation is formulated as follows:

$$\begin{aligned} g_0 \times_{GF(2)} X_{n-1} &= y_{2n-3}, \\ g_1 \times_{GF(2)} X_{n-1} &= y_{2n-2}, \\ g_0 \times_{GF(2)} X_n &= y_{2n-1}, \\ g_1 \times_{GF(2)} X_n &= y_{2n}. \end{aligned} \tag{2}$$

However, we find this situation does not happen in the whole extra bits determination process when only one ZigBee channel is considered, as the deinterleaving process has scattered the significant bits far way enough to avoid this situation, no matter in which combination of QAM modulations and ZigBee channels. Thus, the *twin significant bits* can be satisfied through inserting two extra bits to the WiFi data bits in the designated positions.

The transmit bits $\{x_n\}(n \in [1, N])$ can be generated through inserting extra bits to WiFi data bits $\{x'_i\}$. We formulate the general process in Algorithm 1. Please note that both $\{x'_i\}$ and $\{x_n\}$ are the scrambled bits. The final transmit bits will be obtained through descrambling $\{x_n\}$. From the first bit in $\{x'_i\}$, the device determines whether it triggers a significant bit. If yes, it calculates the extra bits etr_0 or etr_1, then adjusts the values of $\{x_n\}$; if not, it simply assigns current x'_i to x_n. The process is conducted until all the data bits $\{x'_i\}$ are traversed. Algorithm 1 is a general process for the 1/2-coding rate; for any other coding rate, there will be another general transmit bits generation process due to different values and positions of extra bits. We do not cover all the coding rates here as the basic process is similar.

E. Impact of Parallel ZigBee Transmissions

From the aforementioned analysis, we see that the significant bits can be completely satisfied when considering only one ZigBee channel. However, when considering parallel ZigBee transmissions in multiple channels, the situation is more complicated. This part investigates the impact of parallel ZigBee transmissions.

We study the scenario of ZigBee parallel transmissions in two channels, the results are shown in Table III. We see that in most cases the significant bits can be satisfied completely (see the '✓' in Table III); that means, the QAM points in the overlapped subcarriers of the two channels are all with the lowest power.

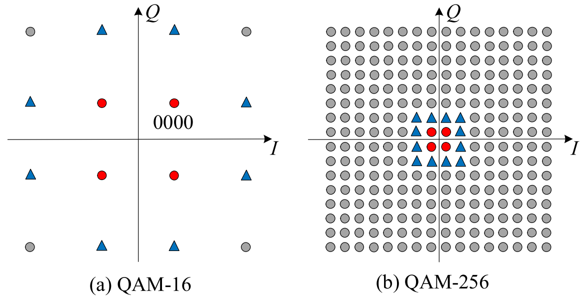


Fig. 11. Illustration of significant bit mismatch under QAM-16 and QAM-256.

However, there are still some cases that the significant bits cannot be satisfied completely (see the '×' in Table III), such as the case that ZigBee uses CH1 and CH4 while WiFi adopts QAM-16, as the abnormal situation of (2) occurs in these cases.

We go through the abnormal situations of (2) in all the cases, and find that there is always a *single significant bit* among y_{2n-3} and y_{2n-2}, while there are always *twin significant bits* among y_{2n-1} and y_{2n}. Based on this observation, we let y_{2n-1} and y_{2n} be satisfied first when the situation of (2) occurs, leading to only one unsatisfied significant bit (y_{2n-3} or y_{2n-2}) in each situation. The unsatisfied significant bit will turn a QAM point into an adjacent one. Since QAM-64 has no such problem, we only show the cases of QAM-16 and QAM-256 in Fig. 11, where the red circles indicate the ideal QAM points with the lowest power, and the blue triangles indicate the adjacent QAM points due to unsatisfied significant bits. We see that this situation obviously has much higher impact on the ZigBee performance when QAM-16 is adopted, as the power of adjacent points is very close to the averaged signal power; under QAM-256, this impact is very small as the adjacent points still have very low power.

Algorithm 1 can also be used in this scenario to generate the transmit bits $\{x_n\}$, according to the required significant bits $\{v_k, p_k\}$ after convolutional encoder and the original data bits $\{x'_i\}$. However, when the abnormal situation of (2) occurs, the added extra bits cannot make the required significant bits $\{v_k, p_k\}$ satisfied; that means, when the transmit bits $\{x_n\}$ are passed through convolutional encoder, the significant bits in some subcarriers are inconsistent with $\{v_k, p_k\}$, making the QAM points not the ideal ones (the red circles in Fig. 11), but the adjacent ones (the blue triangles in Fig. 11). We should point out that, among the seven subcarriers within each ZigBee channel, the unsatisfied significant bits always occur in only one subcarrier, while the other subcarriers are filled with QAM points with the lowest power, making SledZig still with high performance in this situation. Here we do not analyze the scenarios of three and four parallel ZigBee channels as SledZig is not recommended to use in these scenarios due to the increased WiFi throughput loss.

F. Impact of Pilot

Each ZigBee channel in CH1-CH3 overlaps with a pilot subcarrier. Since the pilot subcarrier has much higher power than the data subcarriers with the lowest power, it obviously deteriorates the performance of SledZig since the averaged signal power at ZigBee is increased.

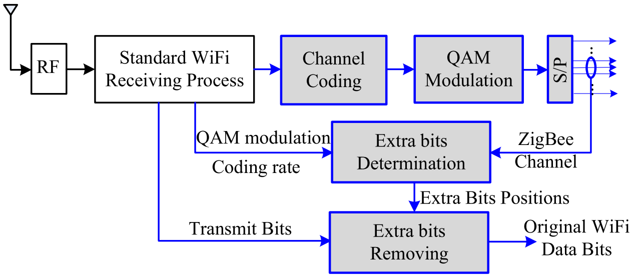


Fig. 12. Processs at the WiFi receiver.

In addition, one may argue that, although the averaged signal power at the ZigBee channel decreases, the high power within this short channel band would have much stronger interference to Zigbee, making its transmission unsuccessful. However, the DSSS modulation adopted by ZigBee can naturally tolerate this kind of interference. DSSS makes the transmitted signal wider in bandwidth than the original data bandwidth. If part of the transmission is corrupted, the data can still be recovered from the remaining part of the signal. Thus, as long as the WiFi signal can be decreased to make the ZigBee SNR (signal to noise ratio) meet the requirements of decoding, the ZigBee transmission can be successful.

G. Process at the WiFi Receiver

The process at the WiFi receiver side is quite simple, as shown in Fig. 12. The receiver first conducts the standard WiFi receiving process to obtain the transmit bits, then removes the extra bits to get the original WiFi data bits. The positions of the extra bits are fixed in the transmit bits, and they are determined by three kinds of information: the ZigBee channel, QAM modulation and coding rate. The latter two pieces of information can be obtained directly from the PLCP (physical layer convergence protocol) header of the WiFi packet [18]. The key issue here is to obtain the ZigBee channel. With the transmit bits, the WiFi receiver can conduct the channel coding, QAM modulation and S/P processes shown as the grey blocks in Fig. 12, which are exactly the same as the blocks in Fig. 4; it can then observe the QAM points and determine the ZigBee channel: the QAM points on the overlapped subcarriers are all lowest ones. This process is fully compatible with the 802.11 standard.

VII. Combating WiFi Preamble Interference

The previous design only decreases the signal power of the WiFi payload. However, the WiFi preamble can still affect the ZigBee transmission. In this part, we first investigate the impact of WiFi preamble, then make quantitative analysis of WiFi preamble interference, and finally design an interference-resistance mechanism on ZigBee to recover the collided ZigBee packet under this situation.

A. Impact of WiFi Preamble

Each WiFi packet includes a preamble for synchronization and CFO (central frequency offset) estimation, as shown in

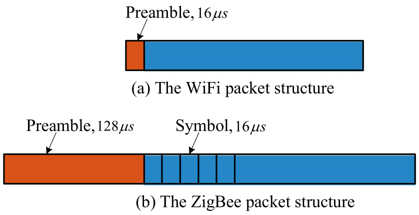

Preamble, 16μs
(a) The WiFi packet structure
Preamble, 128μs
Symbol, 16μs
(b) The ZigBee packet structure

Fig. 13. Packet structure.

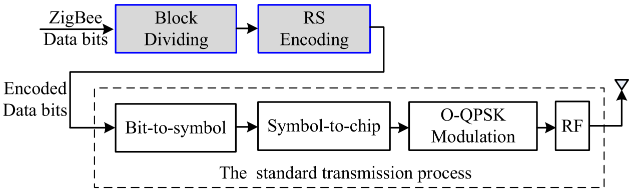


Fig. 14. ZigBee transmission process with RS encoding.

Fig. 13(a). The preamble contains 10 repetitive STS (short training symbols) and two repetitive LTS (long training symbols); it lasts for 16 μs in total. Meanwhile, each ZigBee packet includes a preamble which lasts for 128 μs before the payload; the payload contains a set of ZigBee symbols to carry data bits, while each ZigBee symbol lasts for 16 μs, as shown in Fig. 13(b). We analyze the impact of WiFi preamble from the two scenarios shown in Fig. 4.

For the scenario of Fig. 4(a) where SledZig decreases the WiFi carrier sense range to enable more ZigBee transmissions, the impact is negligible. The ZigBee CCA period must be eight symbols [19], that is 128μs. Thus, in case the WiFi preamble is within a ZigBee CCA period, this 16μs high power signal has very limited impact on the CCA result, comparing with the 112μs low power signal.

For the scenario of Fig. 4(b) where SledZig reduces the WiFi interference to ZigBee transmission, the impact is more complicated. In case the WiFi preamble interferes with the ZigBee preamble, this sudden interference will not affect the detection of ZigBee preamble due to its redundancy design. However, in case the WiFi preamble interferes with a ZigBee symbol in the payload, this symbol will not be detected correctly with a high probability.

In the following parts, we analyze this WiFi preamble interference and design an interference-resistance mechanism to combat it.

B. Quantitative Analysis of WiFi Preamble Interference

A WiFi preamble may interfere with one or two ZigBee symbols, as both the WiFi preamble and ZigBee symbol last for 16 μs, as shown in Fig. 13. However, ZigBee adopts DSSS in the physical layer, which provides the device with the ability to combat this interference to a certain extent.

The standard ZigBee transmission process is shown as the white blocks in Fig. 14. Every four data bits are mapped into one data symbol, and each symbol is mapped into a 32-chip PN sequence through DSSS. The chip sequence representing each data symbol is modulated into I/Q phase samples using O-QPSK; that means, the odd chips are modulated onto I phase and the even chips are modulated on to Q phase. The samples are finally transmitted after the RF front end. Each chip lasts for 1μs. The data symbol can be detected correctly even when parts of the 32 chips are interfered, according to the DSSS design. Thus, whether a WiFi preamble can interfere with a ZigBee data symbol is determined by how many chips in this symbol are interfered.

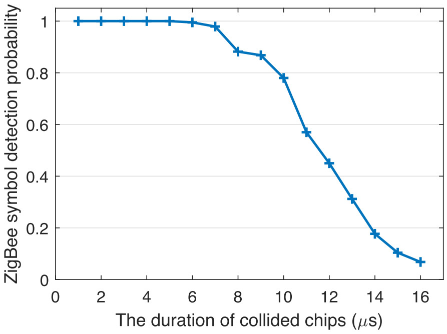


Fig. 15. ZigBee symbol detection probability under collided chips.

We conduct simulations to analyze how a WiFi preamble interferes with a ZigBee data symbol quantitatively, the results are shown in Fig. 15. We see that when a WiFi preamble overlaps with a ZigBee symbol by no more than 12chips (corresponding to 6 μs), the four data bits carried by this symbol can be detected successfully with the probability of 100%. When the number of collided chips increases to 20 (corresponding to 10 μs), the symbol detection probability decreases to less than 80%. Thus, we have the conclusion that a WiFi preamble can affect the reception of one ZigBee symbol with a high probability, and affect two adjacent two ZigBee symbols with a small probability.

Besides the aforementioned analysis, we should also analyze how many WiFi packets would interfere with one ZigBee packet. A ZigBee packet duration is no more than 4 ms. Considering that many WiFi packet lasts for several hundreds of microseconds, a ZigBee packet has a high probability to be interfered by more than one WiFi packet, that means, it may be interfered by multiple preambles with long intervals.

In the wireless communication process, this kind of burst interference with a small amount is very suitable for error correction through channel coding. Based on the previous analysis, we propose an interference resistance mechanism in the following part.

C. Interference Resistance Design

In this part, we adopt the Reed-Solomon (RS) codes to encode the ZigBee data bits, so as to combat the WiFi preamble interference.

RS codes are powerful technique for error detection and correction, through adding redundancy bits to the data bits. RS codes operate on a block of data, which contain a set of Galois-Field (GF) elements called RS symbols. RS codes are always represented as RS(n, k), where n represents the total number of symbols in one code block, k is the number of information symbols in that block, while $t = n - k$ is the number of check symbols used to correct up to $t/2$ erroneous symbols. We note

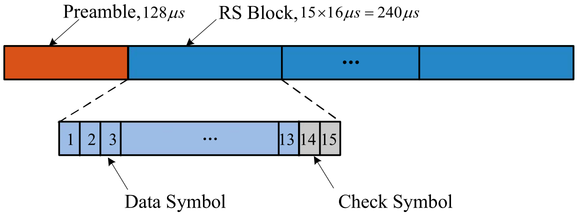


Fig. 16. Illustration of RS encoding on the ZigBee data bits.

that the symbol n here has different meaning with that symbol n used in Section VI-D.

When the RS codes are applied to this context, we need to choose some key parameters. At first, we choose to use the Galois-Field $GF(2^4)$ to make each element in the field contain four bits, so that a RS symbol naturally corresponds to a ZigBee data symbol, making the processes at both the ZigBee transmitter and receiver simple. In this context, we let $n = 15$, that means, each RS block contains 15 ZigBee symbols, corresponding to $15 \times 16\mu s = 240\mu s$, as shown in Fig. 16. Considering that the duration of WiFi packets always ranges from hundreds of microseconds to a few milliseconds, we can conclude that each RS block may be interfered by one WiFi preamble, but has little possibility to be interfered by more than one WiFi preamble. Combining the investigation of DSSS in Fig. 15, we may set $t = 2$ to correct one erroneous symbol, or set $t = 4$ to correct up to two erroneous symbols. Obviously, the latter can ensure successful data transmission better, but would induce more transmission overhead. Theoretically, RS codes with $t = 2$ and $t = 4$ will lead to about 8% and 16% throughput loss, respectively. We will make further study for the performance through hardware experiments in Section VIII.

The process at the ZigBee device is relatively simple. As shown in Fig. 14, the data bits are first divided into a group of blocks; each block has $4 \times k$ data bits (if the last block is not long enough, '0' will be added); after RS encoding, $15 - k$ check symbols are added, making each block with 4×15-bit length; the encoded data bits containing a set of RS blocks are then fed into the standard ZigBee transmission process. The ZigBee receiver first goes through the standard process to obtain the encoded data bits in the payload, then divides them into a group of blocks with 4×15-bit length; it finally conducts RS decoding for each block and obtains the original data bits. This process is compatible with the ZigBee standard when the parameter t is pre-determined. We note that it cannot achieve flexible RS code selection according to the network situations, which involves information exchange between the transmitter and receiver. We leave it as our future work.

VIII. Experiments

In this section, we evaluate the performance of SledZig through hardware experiments.

A. Experimental Setup

We implement a prototype of SledZig based on USRP (universal software radio peripheral) N210 and TelosB. WiFi is implemented by USRP N210 with the open-source GNURadio for signal processing. ZigBee is implemented on TelosB, a well-known ZigBee platform on the 2.4 GHz band. As shown in Fig. 17, we use one USRP as the WiFi transmitter (WiFi Tx) to generate the WiFi signals following the IEEE 802.11 standard, and use another USRP as the WiFi receiver (WiFi Rx). For a WiFi packet, we first insert extra bits to it according to the SledZig design to generate the transmit bits, then feed the transmit bits to the WiFi transmission process in WiFi Tx to generate the required signal. We use two TelosB devices as the ZigBee Tx and Rx to test the ZigBee performance.

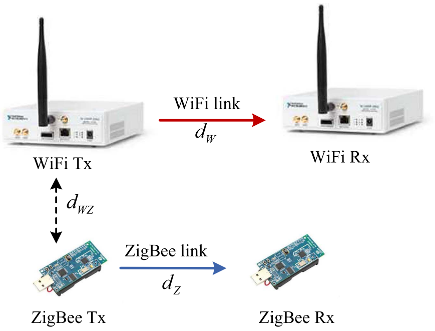


Fig. 17. Experimental setup.

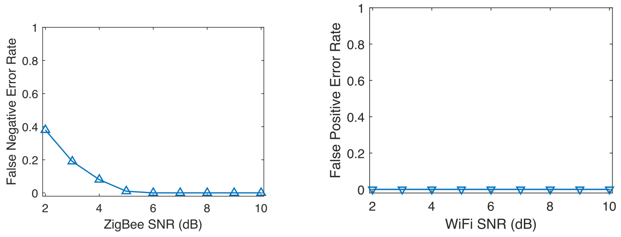
Experiments are conducted in a $10m \times 15\,m$ open space office. The background noise is tested to be $-91\,dB$. The USRP Tx and Rx work at the $13th$ WiFi channel. The two TelosB devices work at the four overlapped ZigBee channels numbered from 23 to 26. Here the ZigBee channels 23-25 are CH1-CH3, and the channel 26 is CH4. Since a WiFi channel overlaps with four ZigBee channels in the same pattern, the performance investigated in this WiFi channel can also represent the performance in other channels.

For the easy of description in the following parts, we denote the distance between the WiFi and ZigBee links as d_{WZ}, denote the link distance between WiFi Tx and Rx as d_W, and denote the link distance between ZigBee Tx and Rx as d_Z, as shown in Fig. 17.

B. ZigBee Channel Identification

In this experiment, we use two metrics to measure the ZigBee channel identification performance: false negative error rate which indicates the probability that the channel of a ZigBee signal cannot be identified correctly, and false positive error rate which indicates the probability that a WiFi signal is mistakenly identified as a ZigBee signal on a certain channel.

We first test the identification performance for ZigBee signals. We collect ZigBee signals from four ZigBee channels (CH1-CH4) under different SNRs. The specific SNR is obtained through adjusting the ZigBee transmission gain. We collect 100 packets for each configuration, and use 2,560 samples for each packet to conduct the ZigBee channel identification. As the performance under each ZigBee channel has little difference, we just show the averaged false negative error rates in Fig. 18(a). We see that when the ZigBee SNR is higher than $6\,dB$, the false negative error rate is nearly zero, which means the ZigBee



(a) False Negative Error. (b) False Positive Error.

Fig. 18. ZigBee channel identification error.

channel can be identified correctly with the probability of 100%. The error rate is as high as 8.2% even when SNR is 5 dB, which is over the threshold β_Z. We consider it due to the signal energy fluctuation across the four subchannels. The error rate is about 38% when SNR is 2 dB due to the high setting of β_Z.

We then test the identification performance for WiFi and Bluetooth signals. We collect WiFi signals under different SNRs, and the specific SNR is obtained through adjusting the WiFi transmission gain. We collect Bluetooth signals based on commercial devices, while the SNR is adjusted through changing the transmitter-receiver distance. We collect 100 packets for each configuration and use 2,560 samples for each packet to conduct the process. The results are shown in Fig. 18(b). We see that the false positive error rate is nearly zeros in all SNR situations, which means the WiFi or Bluetooth signal is impossible to be identified as a ZigBee signal. We notice that the error rate is still about 100% when the WiFi SNR is less than 4 dB, as the WiFi signal power difference in each 1 MHz band is also lower than β_Z in this situation.

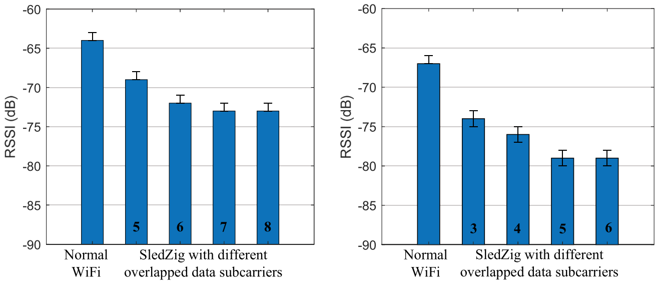
Fig. 18 just gives the error rates under the single ZigBee channel scenario. Actually, the performance under parallel ZigBee channel scenarios is almost unchanged, as two ZigBee channels are far away enough with no mutual interference and each channel is identified independently.

C. *ZigBee Performance*

The main objective of this paper is to decrease the WiFi signal power in the ZigBee channel to improve the ZigBee network performance, through both avoiding interference and exploiting transmission opportunities. Here we conduct experiments to quantify the performance. As the main advantage of SledZig compared with the previous works is that it is fully standard-compatible, here we only compare the performance of SledZig with standard using normal WiFi signals.

1) RSSI at ZigBee: TelosB uses RSSI (received signal strength indication) to measure the received signal power. Since the SledZig design is to decrease the WiFi signal power on the ZigBee channel, this leads to a lower RSSI at ZigBee compared to the standard WiFi signal. Actually, how much RSSI can be reduced will finally affect how much ZigBee performance can be improved. We first investigate RSSI based on the prototype.

According to the theoretical analysis in Section VI-B, the optimal number of overlapped data subcarriers with a ZigBee channel is seven for CH1 to CH3, and five for CH4. We test it through experiments. Here the distance between WiFi Tx and ZigBee Rx is fixed at 1 m, and the transmission gain of WiFi Tx is 15. Fig. 19 shows the collected RSSI in four ZigBee channels under QAM-64 as an example. Due to the varied environment and the limitation on the hardware testbed, the collected RSSI under the same situation is not fixed but has $1 \sim 3\,dB$ variation. We see that in CH1-CH3, the RSSI with seven data subcarriers is about $1 \sim 2\,dB$ lower than that with six subcarriers, and it remains unchanged when the number of subcarriers increases to eight. We also see that five data subcarriers are suitable for CH4. Besides that, the RSSI from SledZig signal with QAM-64 has about 7 dB decrease in CH1-CH3, and about 12 dB decrease in CH4, comparing with the normal WiFi signal where the transmit bits is the randomly generated data bits.

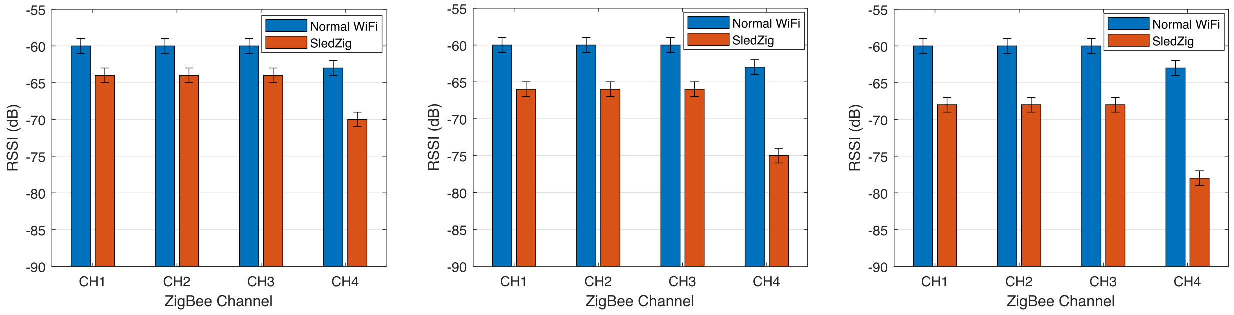


(a) CH1-CH3 (b) CH4

Fig. 19. Impact of the number of data subcarriers on RSSI at ZigBee.

We then conduct experiments to investigate the decrease of RSSI under different QAM modulations and ZigBee channels, the results are shown in Fig. 20. We note that the RSSI from normal WiFi signal has little change when the QAM modulation varies due to the similar averaged signal power. Meanwhile, RSSI collected on CH1, CH2 and CH3 nearly remains unchanged, because the three channels have the similar feature: they are all overlapped with one pilot and seven data subcarriers. In addition, RSSI collected on CH4 is about $3 \sim 4\,dB$ lower than that on CH1-CH3, since there are two null subcarriers with no power in CH4. In CH1-CH3, SledZig can decrease RSSI from about $-60\,dB$ to $-64\,dB$ under QAM-16, to $-66\,dB$ under QAM-64, and to $-68\,dB$ under QAM-256. The situation in CH4 is much better, RSSI can be decreased from about $-63\,dB$ to $-70\,dB$ under QAM-16, to $-75\,dB$ under QAM-64, and to $-78\,dB$ under QAM-256. That is because the pilot subcarrier in CH1$\sim$CH3 can largely increase the averaged signal power. From these results, we see that a ZigBee network can have the highest performance when it works on CH4.

2) ZigBee Throughput Without Interference: Before investigating the ZigBee performance under interference, we first figure out the ZigBee performance without interference as a reference. We let the WiFi Tx not transmit packets, but let the ZigBee Tx transmit packets continuously. The TelosB transmission gain (Tx gain) can be set from 0 to 31, while 31 is the maximum gain and corresponds to the maximum transmission power. We conduct experiments to investigate the ZigBee power level in terms of the link distance d_Z and Tx gain. As shown in Fig. 21, we see that even when d_Z is 0.5m, the RSSI is only about $-75\,dB$ under the maximum transmission power (Tx gain is 31). When d_Z is 1 m and Tx gain is below 15, the signal is submerged in background



(a) QAM-16. (b) QAM-64. (c) QAM-256.

Fig. 20. RSSI collected at ZigBee under each channel and modulation type.

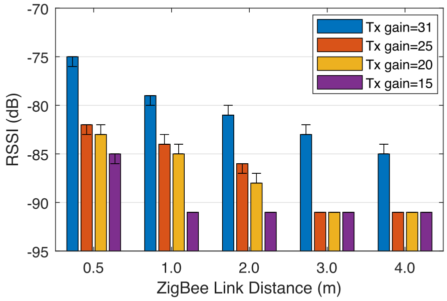
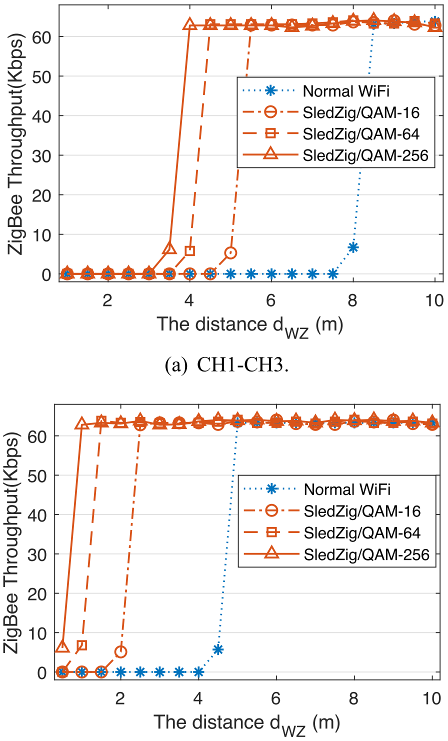


Fig. 21. RSSI in terms of ZigBee link distance d_Z and Tx gain.



(a) CH1-CH3.

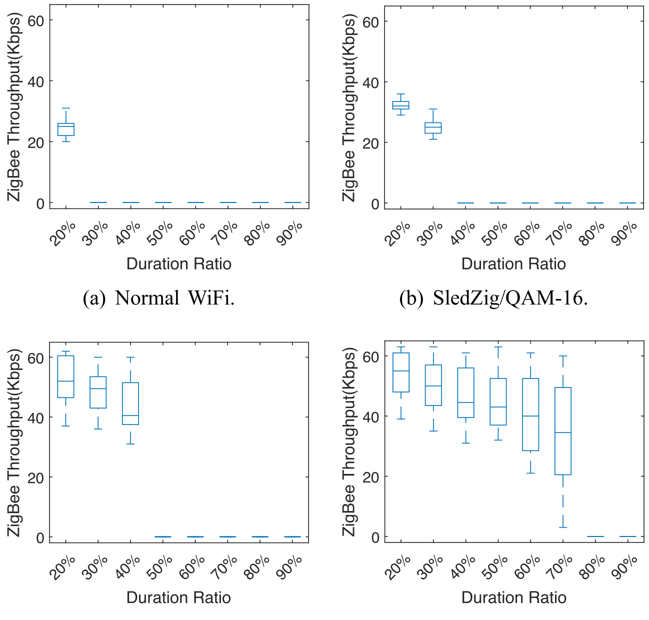
(b) CH4.

Fig. 22. ZigBee throughput in terms of d_{WZ} under continuous WiFi transmission.

noise, that is $-91\,dB$. When d_Z is $3\,m$ or larger, the collected RSSI decreases to the background noise even when Tx gain is 25. In addition, the ZigBee throughput without interference is about $63\,Kbps$, which is much lower than the $250\,Kbps$ data rate in the PHY layer. This result coincides with many previous works such as [3]. It mainly comes from CSMA/CA in the MAC layer design. As described in Section II-B2, before transmitting a data packet, the device should perform CCA during DIFS and backoff process to determine the channel is idle or busy, and only when the channel is determined idle during DIFS and all the backoff time slots can the device transmit the data packet. Therefore, the obtained ZigBee application layer data rate will be significantly reduced compared to the physical layer even without interference.

3) Impact of d_{WZ}: We then evaluate the ZigBee performance under continuous WiFi transmissions. That means, the WiFi packets are transmitted without any interval and the ZigBee transmission will certainly be collided by the WiFi transmission. The ZigBee performance in terms of d_{WZ} is shown in Fig. 17. The WiFi Tx gain is set to be $15dB$. The link distance d_Z is set to be $1\,m$. Fig. 22 shows the ZigBee throughput of SledZig under three QAM modulations compared with normal WiFi. We see that with SledZig, the ZigBee transmission can be successful when the Zigbee link is closer to the WiFi link. Specifically, for ZigBee link in the channels of CH1-CH3, the ZigBee throughput can be about $63\,Kbps$ under normal WiFi interference only when d_{WZ} is at least $8.5\,m$, while this distance can be shortened to about $3.5\,m$, $4.5\,m$ and $5\,m$ with SledZig under QAM-256, QAM-64 and QAM-16 respectively, because the WiFi signal power in the channel can be largely reduced by SledZig. The situation is a little different in CH4, as the overall WiFi signal power in this channel is about $4\,dB$ lower than that in CH1-CH3. We see from Fig. 22(b) that SledZig can make Zigbee transmission successful under QAM-256 even when d_{WZ} is as short as $1\,m$. When the Tx gain increases or decreases, the ZigBee throughput varies, but the general trend does not change. With SledZig, ZigBee links which are nearer the WiFi transmitter have more opportunities to transmit packets successfully. The main reason is that the decreased WiFi signal power shortens the WiFi carrier sense range for ZigBee (d_{CS}^W in Fig. 5(a)).

4) Impact of WiFi Traffic: The previous experiments are conducted under continuous WiFi transmissions. Actually, when the WiFi data rate decreases, the ZigBee throughput can be



(a) Normal WiFi.

(b) SledZig/QAM-16.

(c) SledZig/QAM-64.

(d) SledZig/QAM-256.

Fig. 23. ZigBee throughput under different WiFi data traffic.

further improved. In Fig. 22(a) we see that, when the distance d_{WZ} is less than $3\,m$ in CH3, all the mechanisms have very poor performance under continuously WiFi transmission. We then conduct experiments to investigate the impact of WiFi data traffic. We fix d_{WZ} to be $1\,m$, fix d_Z to be $0.5m$, where the ZigBee link has high probability to be interfered by the WiFi signal according to the tested RSSI. We change the parameter of duration ratio to measure the ZigBee performance in this situation. The duration ratio is defined as the ratio of the WiFi data transmission duration in the channel. The value represents the amount of data traffic in the application layer. We change the ratio from 20% to 90%, making the WiFi traffic increase gradually. We see a variation in the ZigBee performance within a certain range in the experimental results, because how a ZigBee packet is interfered by a WiFi packet is random in the experiments due to the random backoff time slots in CSMA/CA; that means, some ZigBee packets are not interfered, some packets' preambles are interfered, while some other packets' payload are interfered. Thus, we use box plots to show the results, as depicted in Fig. 23. We see that SledZig can improve ZigBee throughput significantly under lower data traffic. The throughput under normal WiFi interference is only about $23\,Kbps$ when the ratio is 20%, and it is nearly zero when the ratio increases. However, SledZig has high throughput even when the ratio is 70% under QAM-256, 40% under QAM-64 and 20% under QAM-16. Specifically, the average throughput is $34.5\,Kbps$ when the ratio is 70% under QAM-256, while the lower quartile can still be about $20\,Kbps$.

We further test the performance of SledZig under actual WiFi network traffic. We first use an USRP to collect data on a WiFi channel and assess the channel occupation under three typical situations: (1) the regular situation when WiFi is used in an office, (2) watching video through WiFi, and (3) downloading files with high speed. The observed WiFi channel is 11, corresponding to the central frequency of $2.462\,GHz$. We find that the WiFi occupation rates (duration ratio) under the three situations are about 19.1%, 22.1% and 71.2%, respectively. We then let USRP transmit WiFi packets with the three kinds of duration ratios under normal WiFi and SledZig, and the tested ZigBee performance is shown in Fig. 24. We see that SledZig under the three QAM modulations can increase the ZigBee throughput significantly in the first two situations. However, under the situation of downloading files, only SledZig with QAM-256 can achieve an averaged $33.6\,Kbps$ ZigBee throughput with very large variance, due to the high duration ratio.

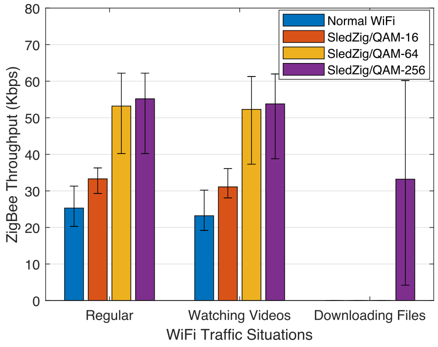
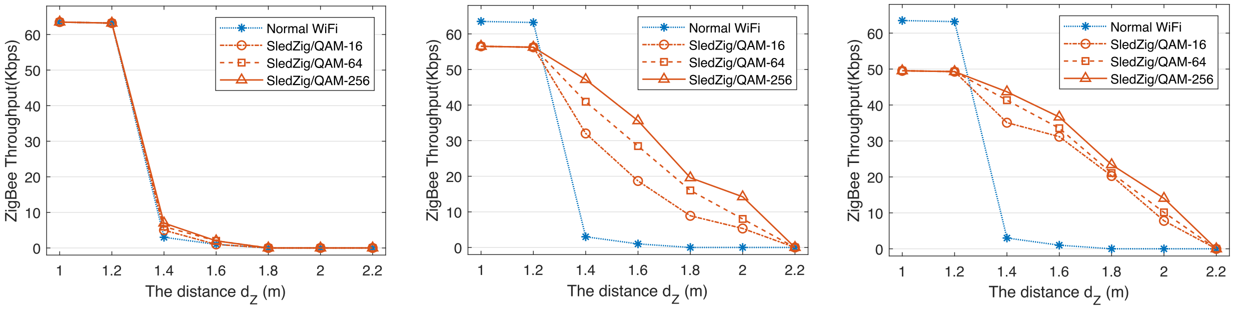


Fig. 24. ZigBee throughput under different WiFi traffic situations.

5) Impact of d_Z: We should note that the ZigBee performance improvement in the previous experiments comes from more transmission opportunities, since the decreased WiFi signal power shortens the WiFi carrier sense range for ZigBee (d^W_{CS}, as shown in Fig. 5(a)). In this part, we intend to figure out how much ZigBee performance can be improved through decreasing interference by SledZig, as the scenario shown in Fig. 5(b). We use the ZigBee channel of CH3, and set d_{WZ} to be $8.5\,m$ to make ZigBee Tx have the opportunity to transmit packets even under the normal WiFi signal. We set the ZigBee transmission gain to 25, and then change the distance d_Z from $1.0\,m$ to $2.5\,m$ through moving the ZigBee receiver to test the ZigBee throughput. We let the WiFi device transmit signals continuously. That means, this experiment is under continuous WiFi transmissions and the ZigBee payload inevitably overlaps with the WiFi preamble.

The results are shown in Fig. 25. We see that when d_Z decreases to $1.4\,m$ under normal WiFi transmission, the ZigBee throughput is nearly zero, as the ZigBee signal is too weak compared to the WiFi signal, making SINR (signal to interference and noise ratio) below the required threshold. We also see from Fig. 25(a) that SledZig without RS encoding brings little throughput improvement in this case even under QAM-256 due to the high power of WiFi preamble. However, with RS encoding, the Zigbee performance can be improved significantly when d_Z is below $2.2\,m$. Specifically, RS encoding with four check symbols leads to higher performance than RS encoding with only two check symbols, as one WiFi preamble can interfere with two ZigBee symbols in some cases. We see that the maximum ZigBee throughput decreases from $63\,Kbps$ to about $56.5\,Kbps$ in Fig. 25(b), and to about $50\,Kbps$ in Fig. 25(c), due to the overhead induced by RS encoding. However, this



(a) No RS encoding. (b) RS encoding with two check symbols. (c) RS encoding with four check symbols.

Fig. 25. ZigBee throughput in terms of d_Z under continuous WiFi transmission.

TABLE IV
WiFi Throughput Loss Under Different Settings

Modulation	Coding Rate	No. of bits per OFDM symbol	No. of extra bits (CH1-CH3)	No. of extra bits (CH4)	Throughput Loss (CH1-CH3)	Throughput Loss (CH4)
QAM-16	1/2	96	14	10	14.58%	10.42%
	3/4	144	14	10	9.72%	6.94%
QAM-64	2/3	192	28	20	14.58%	10.42%
	3/4	216	28	20	12.96%	9.26%
	5/6	240	28	20	11.67%	8.33%
QAM-256	3/4	288	42	30	14.58%	11.72%
	5/6	320	42	30	13.12%	9.37%

design still increases the ZigBee performance significantly as the throughput without it is nearly zero.

D. WiFi Performance

1) Throughput Loss: SledZig requires the WiFi transmitter insert some extra bits to the original WiFi data bits, this process will obviously affect the WiFi throughput. We first make analysis on it.

According to the 802.11 standard, there are two coding rates recommended for QAM-16 and QAM-256, and three coding rates recommended for QAM-64. We see from the design in Section VI-D that the number of extra bits is only affected by the QAM modulation and the ZigBee channel, which together determine the positions of significant bits; this number is not affected by the coding rate, because the encoding processes of all the coding rates are based on the 1/2-rate encoding, and other coding rates are achieved through omitting some of the 1/2-rate encoded bits, while the omitted bits have no effect on the significant bits. Since one significant bit corresponds to one extra bit according to the design in Section VI-D, it is easy to calculate the number of extra bits for each OFDM symbol under different combinations of modulation and ZigBee channel, as shown in Table IV. For example, when QAM-16 is adopted and ZigBee works from CH1 to CH3, the number of extra bits per OFDM symbol can be calculated as 7 subcarriers × 2 significant bits/subcarrier, which is 14.

The throughput loss of WiFi data transmission under the combination of three QAM modulations and the possible coding rates is shown in Table IV. It is calculated as the number of extra bits divided by the number of bits per OFDM symbol. We see that the throughput loss ranges from 6.94% to 14.58%. It decreases with the coding rate under each QAM modulation, because the number of WiFi data bits in each OFDM symbol increases while the number of extra bits remains unchanged. Specifically, the situations of QAM-16 with 1/2-rate encoding, QAM-64 with 2/3-rate encoding, and QAM-256 with 3/4-rate encoding under CH1-CH3 have the highest loss of 14.58%, while QAM-16 with 2/3-rate encoding under CH4 has the lowest loss of 6.94%. In general, the throughput loss for CH4 is lower than that for CH1-CH3, due to fewer extra bits.

We note that the WiFi throughput loss caused by extra bits can be calculated theoretically through this way without further experiments. For an actual WiFi packet transmission, the receiver first detects the WiFi packet to obtain the transmit bits, then remove the extra bits to obtain the original WiFi data bits, as shown in Fig. 12. It then calculates the throughput loss as the number of extra bits divided by the number of transmit bits. This process is the same as that of calculating throughput loss for Table IV.

2) BER Analysis: Although SledZig does not change the standard transmission process, it indeed changes the characteristics of WiFi spectrum, as the signal power on the overlapped subcarriers is decreased. Therefore, we conduct experiments to evaluate the impact of this change on WiFi performance. We use BER (Bit Error Rate) which is commonly used in communication systems as the metric for the evaluation.

We let WiFi TX (one USRP) transmits two kinds of data bits, including the normal WiFi data bits and those with payload encoding, and let WiFi RX (another USRP) receives the WiFi signal and demodulates the data bits. We test the BER of this link under three QAM modulations and different SNR situations,

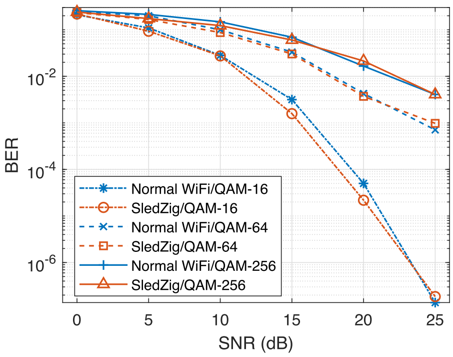


Fig. 26. WiFi performance under standard WiFi and SledZig in terms of SNR.

while the target SNR is obtained through adjusting the WiFi transmission gain. The results are shown in Fig. 26. We see that SledZig has similar BER curve with normal WiFi under the three QAM modulations. Here we note that, theoretically, the BER values under QAM-64 and QAM256 will decrease with the increase of SNR when SNR is higher than $25dB$, but we do not see significant decrease of BER under these situations in the experiments. We consider the result comes from the limitation of USRP. However, we can still see from the fit of the curve that payload encoding has no impact on the BER performance.

Meanwhile, we can also see the big advantage of SledZig over directly reducing WiFi transmission power. For example, under QAM-16, the WiFi BER is 10^{-5} when SNR is $23dB$; at this point, SledZig not only can achieve high WiFi transmission performance, but also can improve ZigBee performance by reducing signal power in the ZigBee channel. The reduced signal power is about $6dB$ in CH4, as shown in Fig. 20. However, if the ZigBee performance is achieved through directly reducing WiFi transmission power by $6dB$, the WiFi BER will be largely increased from 10^{-5} to $10^{-3.5}$. It is more convincing under QAM-64 and QAM256 as more power can be reduced in the ZigBee channel.

3) Impact of ZigBee Interference: According to SledZig design, the decreased WiFi signal power leads to more concurrent ZigBee transmissions. Another question here is, whether the ZigBee transmission can in turn interfere with WiFi data transmission. We then test the WiFi performance under the experiments for Fig. 22 when we evaluate the ZigBee performance under continuous WiFi transmissions in terms of link distance d_{WZ}. Here the WiFi link distance d_W is set to be $3\,m$. We let ZigBee work on CH4 so that the ZigBee device can transmit packets even when d_{WZ} is $2.5\,m$ under QAM-16. We do not test the WiFi BER under QAM-64 or QAM-256 as the USRP cannot provide good performance under the two modulations. The WiFi BER values in terms of d_{WZ} under QAM-16 are shown in Fig. 27. We see that the BER values nearly do no change when d_{WZ} changes, no matter under normal WiFi or SledZig. We consider the main reason for this result is that the ZigBee signal power is about $30\,dB$ lower than the WiFi signal, making it with little impact on the WiFi data transmission.

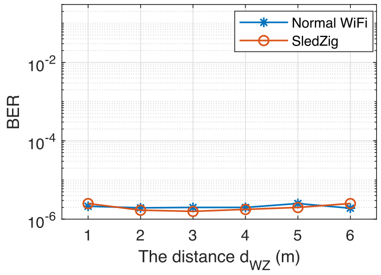
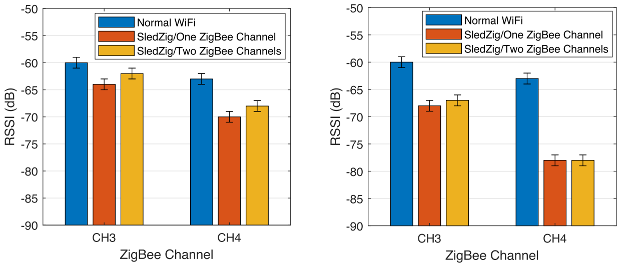


Fig. 27. WiFi BER in terms of d_{WZ}.



(a) QAM-16, 1/2-rate encoding. (b) QAM-256, 3/4-rate encoding.

Fig. 28. RSSI collected at ZigBee under single and parallel ZigBee transmissions.

E. Performance Under Parallel ZigBee Transmissions

In this part, we study the WiFi and ZigBee performance under parallel ZigBee transmissions.

For the WiFi transmission, since more extra bits are required to be inserted to the original WiFi data bits, the WiFi performance will be further degraded. For example, when considering QAM-256 and the ZigBee channels CH1 and CH4, 42 and 30 extra bits should be inserted for each ZigBee channel; thus, the WiFi throughput loss is 26.3% under 3/4-rate encoding, and 22.49% under 5/6-rate encoding. When considering parallel ZigBee transmissions in three or four channels, the WiFi throughput will be even worse.

For the ZigBee transmission, we see from the analysis in Section VI-E that, the significant bits can still be satisfied completely under parallel transmissions when QAM-64 is adopted; thus, the ZigBee performance under each channel in this scenario is the same as that of the single channel scenario. Here we intend to mainly study how this affect the ZigBee peformance under QAM-16 and QAM-256, as the significant bits cannot be satisfied completely in these situations. We study the RSSI collected at ZigBee, as it is a key parameter to measure the performance of SledZig. We do not repeat the other experiments in Section VIII-C due to page limit. We focus on the parallel channels of CH3 and CH4, as shown in Table III. We conduct experiments to make the WiFi TX (one USRP) transmit a WiFi signal, while the transmit bits are generated to lower down the signal power on subcarriers overlapped with both ZigBee CH3 and CH4. The other experimental parameters are the same as those in Section VIII-C1. The results are shown in Fig. 28. We see that under QAM-16 and 1/2-rate encoding, SledZig can only decrease the RSSI by about $2\,dB$ in CH3, by about $5\,dB$ in CH4. Thus, QAM-16 is not recommended in the parallel ZigBee transmission scenario. Under QAM-256, the collected RSSI is not very different from that in the single channel scenario; SledZig can decrease the RSSI by $7dB$ in CH3, by 15 dB in CH4. Actually, in other situations when significant bits cannot be satisfied completely, like CH2$\&amp;$CH3 (referring to Table III), the results are similar with Fig. 28. Therefore, we can still expect a higher ZigBee network performance in parallel ZigBee transmissions when QAM-64 or QAM-256 is adopted.

IX. Related Works

A. Cross-Technology Coexistence

Cross-Technology coexistence has been an important issue for a long time. Existing works can be divided into two categories: interference avoidance and interference resistance.

Interference resistance mechanisms utilize PHY layer solutions to combat CTI. ZIMO [21] separates WiFi and Zigbee signals into different data streams by using the technologies of MIMO (Multiple-Input Multiple-Output) and interference cancellation. CrossZig [4] and PolarScout [12] make ZigBee devices detect the presence of CTI in a corrupted packet and then recover the packet. These schemes always require PHY layer modifications or even new transceiver design, making them incompatible with the standards.

Interference avoidance has attracted much more research interest. Some methods avoid CTI through exchanging coordinated information among heterogeneous devices for protocol design. For example, CBT [22], Weeble [23] and WiCop [24] improve the visibility of ZigBee to WiFi through making ZigBee devices transmit specially designed signals, so that WiFi devices can keep silence during ZigBee transmissions. Gsense [25] makes a WiFi device transmit coordination information to ZigBee devices through a customized preamble, thus to schedule their transmissions. In recent years, some methods utilize the emerging cross-technology communication (CTC) [5], [6], [26], [27], [28] to achieve interference management by enabling explicit coordination between heterogeneous devices [13], [14], [29], [30], [31], [32]. For instance, ECC [13] makes a WiFi AP coordinate data transmissions of all the WiFi and ZigBee devices to avoid interference, thus achieves high network throughput; ECT [29] designs the network layer for CTC and lets a server schedule ZigBee transmissions; Chiron [30] designs a customized gateway to enable concurrent transmissions of WiFi and ZigBee data streams in the same frequency band to reduce the transmission delay; BiCord [32] utilizes bidirectional coordination among heterogeneous devices for efficient RF channel allocation. These mechanisms always induce extra packet transmission and require substantial modifications on the MAC layer mechanism.

Some other methods avoid CTI through making heterogeneous devices working on different frequency bands [3], [15], [33]. For example, G-Bee [33] lets a ZigBee device first identify the 802.11b WiFi channel and then transmit its own data packets on the guard band of WiFi traffic to avoid CTI; it requires all the WiFi devices to work on non-overlapped channels, which is hard to be satisfied in the crowded ISM band. EmBee [3] makes a WiFi device reserve the channel for ZigBee transmission through designing null subcarriers; it requires PHY layer modification as this process is incompatible with the standard WiFi transmission process. By comparison, SledZig can still work in the crowded ISM band without any PHY or MAC layer modifications. Although the payload encoding of SledZig has been discussed in [34], [35], this work makes significant extensions such as ZigBee channel identification and combating WiFi preamble interference, so as to further improve the network performance.

We see from the aforementioned analysis that, all the current works have costs to improve the heterogeneous wireless network performance, such as requiring PHY layer or MAC layer modifications to the standard. The cost in this work is to sacrifice a small portion of WiFi performance to enable ZigBee transmission, which also exists in [3]. We consider that SledZig is less costly compared with the previous works as it is fully standard-compatible and more likely to be deployed to real networks.

B. WiFi Payload Encoding

Recent years have seen some works on designing signals through encoding WiFi payload for data transmission. WEBee [5] designs the WiFi payload to make the WiFi signal emulate a ZigBee signal, which can be detected correctly by a standard ZigBee receiver; BlueFi [6] extends the similar idea to the WiFi-to-Bluetooth scenario; TransFi [8] manipulates the payload of WiFi MIMO (multi-input and multi-output) streams, while the mixed transmitted signals on the air form the emulated signal; these methods use all the WiFi payload for CTC data transmission, although the WiFi channel is $20MHz$ or more but the ZigBee and Bluetooth channels are only $2MHz$ and $1MHz$, respectively. OfdmFi [7] achieves symbol-level energy modulation to deliver CTC information through inserting extra bits to the original WiFi data bits; it analyzes the extra bits insertion from the Viterbi decoding process, and does not give quantitative analysis for the impact when the significant bits are unsatisfied. On the contrary, SledZig makes this quantitative analysis of the extra bits insertion from the convolutional encoding process; it can make the QAM points ideal in the single ZigBee channel scenario, while SLEM [9] cannot achieve this goal although it also makes analysis from the convolutional encoding process. We note that SymBee [36] and BlueFi [37] encode payload to achieve ZigBee-to-WiFi and Bluetooth-to-WiFi CTC transmissions; they work at ZigBee and Bluetooth devices, and the basic idea is totally different from this work.

C. Heterogeneous Signal Identification

Heterogeneous signal identification is an important issue in the coexistence scenario as it assists devices in better channel access decisions. We have seen many previous works focusing on or containing the related research. Some works [38], [39], [40] utilize time-domain and frequency-domain features to identify different kinds of signals. The time-domain features include average on-air time, inter-packet duration, etc; these features are only suitable for a stable environment as they vary rapidly with the device movement and data traffic change. Comparably, the frequency-domain features like frequency spectrum are independent of the environment and are more suitable for signal identification. Some works [41], [42] exploit deep learning for signal modulation or signal type identification through feeding the original samples to a deep learning model. The work [43] classifies heterogeneous signals such as ZigBee on WiFi devices through only analyzing the error patterns on commercial WiFi chipsets. Nearly all the works focus on signal type identification, making them unsuitable for SledZig design. LoFi [16] identifies LoRa channels through physical layer technologies and is in a different context. Embee [3] can identify ZigBee channels through analyzing the central frequency offset of the signal, but it requires the ZigBee preamble to be obtained at first, which limits its application. In this work, we exploit the frequency spectrum feature in the frequency domain to identify the ZigBee signals and corresponding channels. It is easy to implement on devices and has high identification accuracy under various situations.

D. RS Coding in Wireless Networks

Reed-Solomon (RS) are block-based error correcting codes. The number and type of errors that can be corrected depend on the RS code design. RS codes have many applications in digital communications and storage, such as wireless and mobile communications, satellite communications, digital television, storage devices, etc. In the WiFi and ZigBee wireless networks, we have seen many related works on exploiting RS codes to combat interference in various scenarios. For example, authors in [44] investigate using selected RS codes to improve ZigBee communication robustness and reduce power consumption. BuzzBuzz [45] exploits RS code to recover a ZigBee packet interfered by the whole WiFi packet transmission, thus needing more parity code. SafetyNet [17] embeds RS correction bits into the ZigBee physical layer to combat cross-technology interference, thus enhancing the ZigBee transmission robustness. GuardRider [46] utilizes RS code in the WiFi backscatter communication scenarios to improve the quality of service (QoS) of backscatter transmission. RS codes in these mechanisms are different as they are designed for different scenarios. In this paper, we design RS code in ZigBee devices to combat the burst interference from the WiFi preamble with only $16\mu s$. The design is different from previous works due to this specific scenario.

X. Conclusion

In this paper, we propose SledZig to enable coexistence of heterogeneous wireless devices, so as to improve the network performance. SledZig decreases the WiFi signal power on the ZigBee channel through making constellation points on the overlapped subcarriers with the lowest power. It can be achieved through encoding the WiFi payload to generate the transmit bits; when the transmit bits are passed through the WiFi transmission process, the signal power on the ZigBee channel can be decreased naturally. SledZig is fully compatible with WiFi and ZigBee standard in both the PHY and MAC layers, thus has the potential to be deployed to real networks. We implement and evaluate SledZig on hardware testbed, and experimental results show that SledZig can effectively increase ZigBee transmissions and improve its performance over a WiFi channel with as low as 6.94% WiFi throughput loss.

References

[1] Cisco, "Cisco annual internet report (2018–2023) white paper," 2020.

[2] Verified Market Research, "Global ZigBee market size by standards, by application, geographic scope and forecast," 2020.

[3] R. Chen and W. Gao, "Enabling cross-technology coexistence for extremely weak wireless devices," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 253–261.

[4] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy, "CrossZig: Combating cross-technology interference in low-power wireless networks," in *Proc. IEEE/ACM 15th Int. Conf. Inf. Process. Sensor Netw.*, 2016, pp. 1–12.

[5] Z. Li and T. He, "WEBee: Physical-layer cross-technology communication via emulation," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 2–14.

[6] H.-W. Cho and K. G. Shin, "BlueFi: Bluetooth over WiFi," in *Proc. ACM SIGCOMM Conf.*, 2021, pp. 475–487.

[7] P. Gawlowicz, A. Zubow, S. Bayhan, and A. Wolisz, "Punched cards over the air: Cross-technology communication between LTE-U/LAA and WiFi," in *Proc. IEEE 21st Int. Symp. "A World Wirel. Mobile Multimedia Netw."*, 2020, pp. 297–306.

[8] R. Chen and W. Gao, "TransFi: Emulating custom wireless physical layer from commodity WiFi," in *Proc. 20th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2022, pp. 357–370.

[9] J. Yao, X. Zheng, R. Xie, and K. Wu, "Cross-technology communication for heterogeneous wireless devices through symbol-level energy modulation," *IEEE Trans. Mobile Comput.*, vol. 21, no. 11, pp. 3926–3940, Nov. 2022, doi: 10.1109/TMC.2021.3065998.

[10] IEEE Computer Society. 802.11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications–Amendment 4: Enhancements for very high throughput for operation in bands below 6 GHz," 2013.

[11] IEEE Computer Society. 802.11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications–Amendment 1: Enhancements for high-efficiency WLAN," 2021.

[12] C. Shao, H. Park, H. Roh, W. Lee, and H. Kim, "PolarScout: Wi-Fi interference-resilient ZigBee communication via shell-shaping," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1587–1600, Aug. 2020.

[13] Z. Yin, Z. Li, S. M. Kim, and T. He, "Explicit channel coordination via cross-technology communication," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2018, pp. 178–190.

[14] W. Chen, Z. Yin, and T. He, "Global cooperation for heterogeneous networks," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 1014–1023.

[15] Y. Wang, X. Zheng, L. Liu, and H. Ma, "CoHop: Quantitative correlation-based channel hopping for low-power wireless networks," *ACM Trans. Sensor Netw.*, vol. 17, no. 2, pp. 1–29, Jun. 2021.

[16] G. Chen, W. Dong, and J. Lv, "LoFi: Enabling 2.4 GHz LoRa and WiFi coexistence by detecting extremely weak signals," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.

[17] Z. Yin, W. Jiang, R. Liu, S. M. Kim, and T. He, "SafetyNet: Interference protection via transparent PHY layer coding," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 267–277.

[18] IEEE Computer Society. 802.11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput," 2009.

[19] IEEE Computer Society. 802.15.4, "IEEE Standard for low-rate wireless networks," 2016.

[20] S. Biaz and S. Wu, "Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison," in *Proc. IEEE Symp. Comput. Commun.*, 2008, pp. 130–136.

[21] Y. Yan, P. Yang, X. Li, Y. Tao, L. Zhang, and L. You, "ZIMO: Building cross-technology MIMO to harmonize zigbee smog with WiFi flash without intervention," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 465–476.

[22] X. Zhang and G. S. Kang, "Enabling coexistence of heterogeneous wireless systems: Case for ZigBee and WiFi," in *Proc. 12th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2011, Art. no. 6.

[23] R. Radunovic, R. Chandra, and D. Gunawardena, "Weeble: Enabling low-power nodes to coexist with high-power nodes in white space networks," in *Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol.*, 2012, pp. 205–216.

[24] Y. Wang, Q. Wang, Z. Zeng, G. Zheng, and R. Zheng, "WiCop: Engineering WiFi temporal white-spaces for safe operations of wireless body area networks in medical applications," in *Proc. IEEE 32nd Real-Time Syst. Symp.*, 2011, pp. 170–179.

[25] X. Zhang and K. G. Shin, "Gap sense: Lightweight coordination of heterogeneous wireless devices," in *Proc. IEEE Conf. Comput. Commun.*, 2013, pp. 3094–3101.

[26] X. Zheng, D. Xia, X. Guo, L. Liu, Y. He, and H. Ma, "Portal: Transparent cross-technology opportunistic forwarding for low-power wireless networks," in *Proc. 21st Int. Symp. Theory Algorithmic Found. Protocol Des. Mobile Netw. Mobile Comput.*, 2020, pp. 241–250.

[27] H.-W. Cho and K. G. Shin, "FLEW: Fully emulated WiFi," in *Proc. 28th Annu. Int. Conf. Mobile Comput. Netw.*, 2022, pp. 29–41.

[28] H.-W. Cho and K. G. Shin, "Unify: Turning BLE/FSK SoC into WiFi SoC," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, 2023, Art. no. 40.

[29] W. Wang, T. Xie, X. Liu, and T. Zhu, "ECT: Exploiting cross-technology transmission for reducing packet delivery delay in IoT networks," *ACM Trans. Sensor Netw.*, vol. 15, no. 2, Feb. 2019, Art. no. 20.

[30] Y. Li, Z. Chi, X. Liu, and T. Zhu, "Chiron: Concurrent high throughput communication for IoT devices," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2018, pp. 204–216.

[31] 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 *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 577–585.

[32] Z. Yu et al., "BiCord: Bidirectional coordination among coexisting wireless devices," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst.*, 2021, pp. 304–314.

[33] Y. Chae, S. Wang, and S. M. Kim, "Exploiting WiFi guard band for safeguarded ZigBee," in *Proc. 16th ACM Conf. Embedded Netw. Sensor Syst.*, 2018, pp. 172–184.

[34] J. Yao, H. Huang, R. Xie, X. Zheng, and K. Wu, "SledZig: Boosting cross-technology coexistence for low-power wireless devices," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst.*, 2022, pp. 754–764.

[35] J. Yao and K. Wu, *Cross-Technology Coexistence Design for Wireless Networks*. Singapore: Springer, 2023.

[36] S. Wang, S. M. Kim, and T. He, "Symbol-level cross-technology communication via payload encoding," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, 2018, pp. 500–510.

[37] Z. Li and Y. Chen, "BlueFi: Physical-layer cross-technology communication from Bluetooth to WiFi," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst.*, 2020, pp. 399–409.

[38] J. Meng et al., "Smoggy-Link: Fingerprinting interference for predictable wireless concurrency," in *Proc. IEEE 24th Int. Conf. Netw. Protoc.*, 2016, pp. 1–10.

[39] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L. Norden, and P. Gunningberg, "SoNIC: Classifying interference in 802.15.4 sensor networks," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2013, pp. 55–66.

[40] A. Hithnawi, H. Shafagh, and S. Duquennoy, "TIIM: Technology-independent interference mitigation for low-power wireless networks," in *Proc. 14th Int. Conf. Inf. Process. Sensor Netw.*, 2015, pp. 1–12.

[41] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[42] J. Yao, W. Lou, R. Xie, X. Jiao, and K. Wu, "Mitigating cross-technology interference through fast signal identification," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 2521–2534, Feb. 2023, doi: 10.1109/TVT.2022.3213663.

[43] D. Croce, D. Garlisi, F. Giuliano, N. Inzerillo, and I. Tinnirello, "Learning from errors: Detecting cross-technology interference in WiFi networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 2, pp. 347–356, Jun. 2018.

[44] L. Biard and D. Noguet, "Reed-Solomon codes for low power communications," *J. Commun.*, vol. 3, no. 2, pp. 13–21, Apr. 2008.

[45] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving Wi-Fi interference in low power ZigBee networks," in *Proc. 8th ACM Conf. Embedded Netw. Sensor Syst.*, 2010, pp. 309–322.

[46] X. He, W. Jiang, M. Cheng, X. Zhou, P. Yang, and B. Kurkoski, "GuardRider: Reliable WiFi backscatter using Reed-Solomon codes with QoS guarantee," in *Proc. IEEE/ACM 28th Int. Symp. Qual. Service*, 2020, pp. 1–10.

Junmei Yao received the BE degree from the Harbin Institute of Technology, China, in 2003, the ME degree from the Harbin Institute of Technology, China, in 2005, and the PhD degree in computer science from the Hong Kong Polytechnic University, in 2016. She is currently an assistant professor with the College of Computer Science and Software Engineering, Shenzhen University, China. Her research interests include wireless networks, wireless communications, and mobile computing.

Haolang Huang received the BE degree from Guangzhou University, Guangzhou, China, in 2020. He is currently working toward the ME degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include Internet of Things and cross-technology coexistence.

Jiongkun Su received the BS degree from South China Agricultural University, Guangzhou, China, in 2021. He is currently working toward the ME degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include deep learning and cross-technology coexistence.

Ruitao Xie received the BEng degree from the Beijing University of Posts and Telecommunications, in 2008, and the PhD degree in computer science from the City University of Hong Kong, in 2014. She is currently an assistant professor with the College of Computer Science and Software Engineering, Shenzhen University. Her research interests include AI networking and mobile computing, distributed systems, and cloud computing.

Xiaolong Zheng received the BE degree from the Dalian University of Technology, China, in 2011, and the PhD degree from the Hong Kong University of Science and Technology, China, in 2015. He is currently a research associate professor with the School of Computer Science and Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, China. His research interests include Internet of Things, wireless networks, and ubiquitous computing.

Kaishun Wu received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology. Before joining HKUST(GZ) as a full professor with DSA Thrust and IoT Thrust, in 2022, he was a distinguished professor and director of Guangdong Provincial Wireless Big Data and Future Network Engineering Center with Shenzhen University. He is an active researcher with more than 200 papers published on major international academic journals and conferences, as well as more than 100 invention patents, including 9 from the USA. He received the 2012 Hong Kong Young Scientist Award, the 2014 Hong Kong ICT awards: Best Innovation and 2014 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award.