Enabling Cross-Technology Coexistence for Extremely Weak Wireless Devices

Ruirong Chen and Wei Gao
Department of Electrical and Computer Engineering
University of Pittsburgh
{ruc28,weigao}@pitt.edu

***Abstract*—Cross-technology coexistence is crucial to avoid collisions of wireless transmissions and improve the efficiency of spectrum utilization in today's large-scale wireless network systems, especially the Internet of Things. However, existing approaches to cross-technology coexistence incur additional transmission delay and signal processing overhead, which are unaffordable by extremely weak wireless devices such as embedded sensors and computational RFIDs. These schemes hence fail when being applied to emerging application scenarios, such as smart cities and connected healthcare where weak devices play important roles. In this paper, we design and implement EmBee, a new wireless PHY technique that enables cross-technology coexistence at zero cost or performance loss to these extremely weak wireless devices. The basic idea of EmBee is to exploit the diversity of different wireless technologies' spectrum utilization, so as to adaptively reserve occupied spectrum from the strong devices for weak wireless devices' concurrent data transmissions. We have implemented EmBee over custom wireless hardware and evaluated EmBee under different wireless scenarios. Experiment results show that EmBee can effectively support ZigBee transmissions over a fully occupied WiFi channel without causing any extra delay, while only resulting in 10% WiFi throughput loss.**

***Index Terms*—Wireless networks; cross-technology coexistence; spectrum reservation**

I. Introduction

The past decade has witnessed the explosive growth of the wireless device population, especially with the emergence of Internet of Things (IoT) that computerizes and interconnects physical objects [21] for new computing paradigms [14], [13], [15]. The wireless performance in such a large-scale networked system, however, could be seriously degraded when weak IoT devices operate with heterogeneous wireless technologies (e.g., WiFi, ZigBee and Bluetooth) that compete for the same spectrum: the wireless traffic generated by different technologies will collide with each other and result in frequent transmission failures [9], [8]. Hence, there is a pressing need to allow these wireless technologies to coexist in a common spectrum, so as to maximize the efficiency of spectrum utilization without any performance degradation [1], [23], [20], [28], [19].

A lot of research efforts have been made to enable cross-technology coexistence. Early researchers suggest modifying the wireless MAC protocols to avoid collisions at the cost of extra transmission delay, by sensing transmissions from other wireless technologies [5], [26] and only transmit when the channel is idle [10]. Recent schemes on cross-technology communication further improves the efficiency of such coordinated transmissions at the PHY layer [11], [6], [16]. Other techniques, instead, aim to cancel the cross-technology interference by adopting advanced wireless hardware such as MIMO [25]. However, these existing approaches generally fail when being applied to extremely weak wireless devices, such as embedded sensors, computational RFIDs and implanted biomedical devices that are increasingly important in emerging application scenarios such as smart cities and connected healthcare. On one hand, these devices usually have too small form factors to have high battery capacity. They hence cannot afford any extra delay that may miss the limited opportunities for transmission. On the other hand, the extremely restrained amount of local resources at these devices prohibits any extra wireless signal extraction and processing, which are however, required by existing techniques for channel sensing.

In this paper, we aim to bridge the aforementioned gap by proposing *EmBee* (Embedded ZigBee), a new wireless PHY technology that enables cross-technology coexistence without incurring any additional transmission delay or signal processing overhead on extremely weak wireless devices. The key insight of the EmBee design is different wireless technologies' diversity in spectrum utilization, which balance between the channel bandwidth, transmission range and power consumption in different ways. For example, strong wireless devices use WiFi to achieve high channel bandwidth, (>20Mbps) by using complicated modulation schemes, channel coding techniques and higher transmit power. ZigBee, on the other hand, is usually used by weak wireless devices to minimize the power consumption using a narrowband channel (2MHz) with simplified modulations (OQPSK). Motivated by such diversity, EmBee adaptively adjusts the wireless technology operated by strong wireless devices to reserve a minor portion of its spectrum, for the wireless channel operated by weak devices. In this way, EmBee allows any wireless traffic from weak devices to be transmitted concurrently with that from strong devices, without any modification on wireless operations at weak devices.

The design of EmBee, however, is challenging because of *i)* the difficulty of precisely and efficiently recognizing weak ZigBee signal from an incompatible strong WiFi device and *ii)* the possible performance loss at coexisting wireless technologies. First, since WiFi and ZigBee use completely different PHY techniques and MAC frame formats, any ZigBee packet cannot pass the packet detection at WiFi PHY, and it is hence difficult for a WiFi device sense the existence of

This work was supported in part by the National Science Foundation (NSF) under grant number CNS-1826884, CNS-1812399 and CNS-1812407.

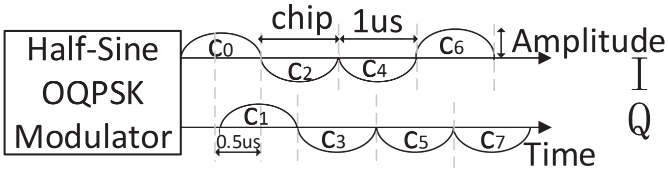


Figure 1: ZigBee data encoding

ZigBee traffic that is simultaneously transmitting in the same spectrum. Second, reserving spectrum from the channel of strong wireless technology will inevitably reduce its channel bandwidth and may weaken the wireless performance provided to the user. How to minimize such performance loss without affecting the weak devices' packet reception is hard.

To precisely detect the weak ZigBee packets from a strong WiFi device, one straightforward approach is to analyze the wireless channel with Fast Fourier Transform (FFT), but it introduces computationally overhead and additional response latency. Instead, EmBee exploits the different frequency offsets in WiFi and ZigBee channels to probabilistically decide whether a wireless signal is produced by WiFi or ZigBee. It further improves the accuracy of such decisions to 93% using statistical channel information, without requiring any coordination across different wireless technologies.

Based on such decisions, EmBee's approach to spectrum reservation builds on the unique properties of modern digital modulation methods, particularly OFDM that will be the technical foundation of next-generation high-speed WiFi [22]: it selectively avoids transmitting WiFi data through some of the OFDM subcarriers, so that the spectrum occupied by these subcarriers becomes available for a concurrent ZigBee channel. Since only a small portion of subcarriers needs to be avoided, EmBee is able to flexibly schedule such spectrum reservation over time, so as to utilize the ZigBee's built-in error correction capabilities for minimum packet reception errors.

We have implemented the EmBee system over customizable wireless hardware, the WARP v3 software-defined radio (SDR) testbed [7], and evaluated the performance of EmBee over commodity ZigBee transceivers. Our experiment results show that EmBee is able to achieve 99% ZigBee packet reception rate over a fully occupied 20MHz WiFi channel, without incurring any extra delay to ZigBee traffic. At the same time, EmBee effectively controls the WiFi throughput loss due to spectrum reservation within 10%, and hence has only negligible impact to most wireless applications.

The rest of this paper is organized as follows. Section II describes how ZigBee works and motivates the EmBee design by explaining the infeasibility of the straightforward FFT-based method for ZigBee detection. Section III provides an overview of EmBee design. Sections IV and V provide technical details about EmBee design on ZigBee detection and spectrum reservation, respectively. Section VI describes EmBee implementation in detail. Section VII evaluates the performance of EmBee in practical wireless scenarios. Section VIII discusses the related work and Section IX concludes the paper.

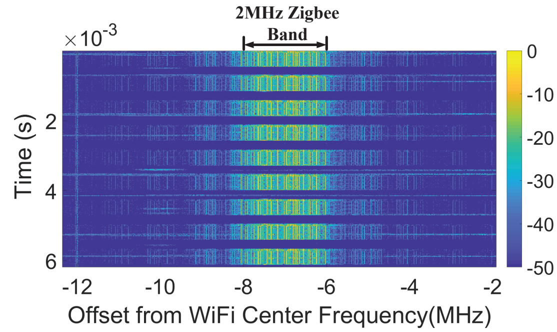


Figure 2: Spectrogram (dB) of ZigBee signal from WiFi

II. Preliminaries

Before describing the EmBee design, in this section we provide the background for ZigBee, as well as motivating the EmBee design by discussing the infeasibility of FFT-based method for detecting ZigBee transmissions. This section provides necessary background regarding the coexistence between Zigbee and WiFi and the difficulties for Zigbee packets to be detected by WiFi.

A. How ZigBee Works?

ZigBee is a low-power wireless technology that is widely used in sensor networks and various types of cyber-physical systems with severe power constraints. To combat against channel interference, the ZigBee PHY uses direct sequence spread spectrum (DSSS) in 16 channels that are separated by 5 MHz with each other. As shown in Figure 1, every ZigBee channel has a bandwidth of 2 MHz encoded by a pseudo-random noise (PN) code, each bit in which is modulated as a chip by orthogonal quadrature phase shift keying (O-QPSK) and transmitted at 2Mchip/s with 1us duration. A ZigBee data symbol, then, contains 4 data bits that are encoded as a PN sequence of 32 chips and results in a data rate of 250 kbps.

B. Why straightforward methods fail?

The most straightforward method of detecting an ongoing ZigBee transmission from WiFi is to harvest the raw RF signal from the ZigBee channel bands and analyze the signal spectrum via FFT. In our preliminary experiment which transmits 1000 ZigBee packets of 84 bytes with -5dBm Tx power (the lowest ZigBee Tx power level), the signal spectrogram shown in Figure 2, which is obtained from a WiFi receiver being placed 2 meters away, demonstrates that the received ZigBee signal strength is at least 20dB higher than the background noise and can be reliably detected by FFT.

However in practice, other types of wireless interference with higher power in the same band could increase the error in detecting the extremely weak Zigbee channel with a single fixed-point FFT. Thus, for satisfiable detection accuracy, such FFT must be performed over a large number of RF samples. For example, when being performed over the WARP v3 testbed, it requires at least 2^{23} RF samples to achieve >95% accuracy, . As shown in in Table I, this computation, even when being performed by a high-performance Virtex-6 FPGA over WARP, results in at least 78.6 ms computational latency that equals to the duration of >20k WiFi symbols, but has to

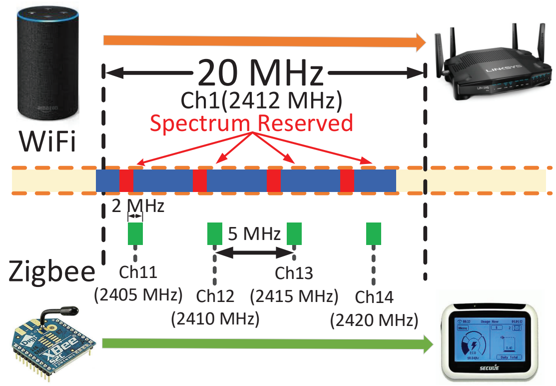


Figure 3: EmBee design

be frequently conducted for timely ZigBee detection and hence incurs serious performance degradation over WiFi devices. Instead in Section IV, we will describe how EmBee ensures accurate ZigBee detection with negligible overhead.

Table I: FFT Overhead

FFT resolution	64	128	256	512
Computational delay (ms)	78.6	79.4	82.4	83.5

III. Overview

The primary objective of EmBee is to allow different wireless technologies to concurrently transmit data over the same spectrum, so as to avoid any extra transmission delay at weak wireless devices without additional computation or explicit coordination. To achieve this objective, EmBee enables a strong wireless device to precisely detect any weak wireless device that is transmitting over a narrowband channel, and then reserves the corresponding spectrum from its wideband channel for concurrent transmissions. For example as shown in Figure 3, dedicated spectrum could be reserved from a WiFi channel for concurrent ZigBee transmissions, without requiring any modification or additional workload on ZigBee devices. Since the bandwidth of a ZigBee channel is 2 MHz and less than 10% of that of a 20 MHz WiFi channel[1], such spectrum reservation only results in minor throughput loss of WiFi.

The performance of EmBee relies on precise detection of an ongoing ZigBee transmission from a WiFi transceiver. To ensure correct spectrum reservation, EmBee requires informations about the existence of an ongoing ZigBee transmission and the specific ZigBee channel being occupied. To minimize the computational overhead, EmBee's approach to such detection builds on the different frequency offsets in WiFi and ZigBee channels. More specifically, it uses the transmission history in the past to collect statistical information about channel characteristics, and further exploits these characteristics to classify every sensed wireless transmission on the air. More details about such detection are described in Section IV.

In theory, EmBee's spectrum reservation will result in at least 15% throughput loss in a 20 MHz WiFi channel, in order to allow one ZigBee device to concurrently transmit. In practice, EmBee strives to further reduce such WiFi throughput loss by exploiting the error correction capability of ZigBee: a ZigBee packet can be successfully decoded even if some of its chips are received as corrupt. As a result, EmBee only needs to reserve the spectrum from a portion of temporally scheduled WiFi symbols, and is also able to flexibly balance between the performance of two wireless technologies by varying such amount of spectrum reservation. Details of such balancing are further described in Section V.

[1] Without loss of generality, similar spectrum reservation can also be applied to WiFi channels with higher bandwidth, such as the 40/80 MHz channels in recent WiFi standards.

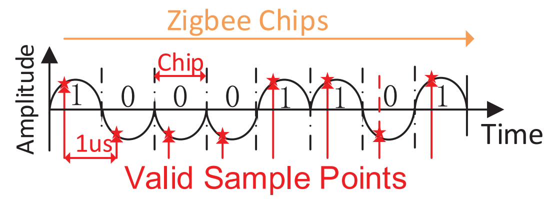


Figure 4: Sampling ZigBee for channel offset calculation

IV. Detecting ZigBee from WiFi

A. WiFi and ZigBee Channel Frequency Offsets

EmBee's primary approach to detecting ZigBee transmissions is to compare the Carrier Frequency Offset (CFO) caused by WiFi and ZigBee transmissions, which can be calculated from raw RF samples with negligible computational overhead. A WiFi system calculates the CFO from its center frequency via auto correlation among the 10 repetitive Short Training Sequences (STS) of WiFi frames [4], each of which contains 16 raw WiFi samples. This calculation method, however, may fail when being directly applied onto a ZigBee transmission, because the WiFi sampling rate (20 MHz) is much higher than the ZigBee chip rate (2 Mchip/s). As a result, a WiFi receiver may sample multiple times over the same ZigBee chip without any periodicity, and these samples hence cannot provide frequency offset information.

To address this challenge, after having detected a sufficiently high amplitude from raw I/Q samples, EmBee starts to sample the ZigBee signal at an enlarged interval that is equal to the ZigBee chip duration, as shown in Figure 4. These samples then correspond to the same position of different ZigBee chips, and the CFO can be computed from these samples, following the same way as the WiFi system operates as:

$$f_O = \frac{1}{16}\angle(\sum_{m=0}^{63} S_m^* S_{m+16}), \tag{1}$$

where S_m indicates a ZigBee sample and S_m^* indicates the complex conjugate of S_m. CFO for Zigbee shows the channel frequency characteristics instead of the offset to WiFi carrier frequency, due to Zigbee chips have no same cyclic property as WiFi STS for CFO computation. On the other hand, To minimize the modifications to current WiFi system, only the ZigBee samples corresponding to the first 5 STSs are being used.

We have experimentally investigated the ZigBee CFO calculated from these samples, by capturing ZigBee signal sent with -5dBm Tx power from a WiFi receiver that is 2 meters away

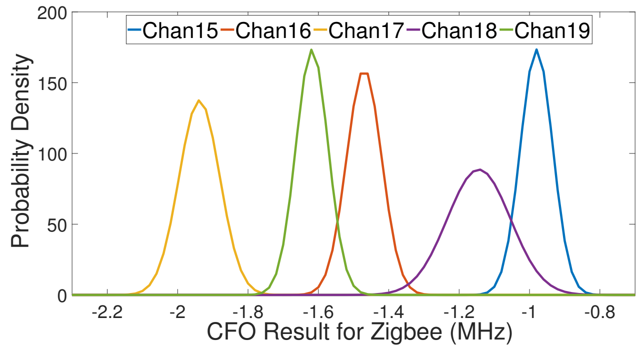


Figure 5: CFO distributions of ZigBee channel 15-19

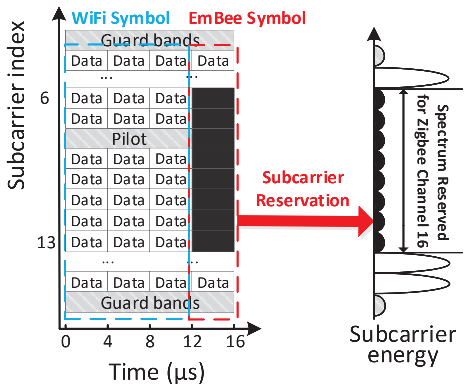


Figure 6: Spectrum reservation via OFDM subcarrier avoidance

and operates at channel 6. As shown in Table II, the CFO of a ZigBee channel, when being calculated from a WiFi device, could be as high as -1.9MHz and at least 500 times higher than that of a WiFi channel (<0.002MHz in most cases). Such difference makes it easy for a WiFi device to distinguish ZigBee transmissions from WiFi traffic based on the calculated channel offset.

Table II: Zigbee CFO calculated from Embee

Zigbee Channel	15	16	17	18	19
Zigbee CFO(MHz)	-0.98	-1.47	-1.94	-1.14	-1.60

B. Probabilistic Decision of ZigBee Channels

Table II also demonstrates that the CFO of different ZigBee channels, when being calculated at WiFi, are significantly different and can be clearly distinguished from each other. However in practice, due to channel noise and multi-path interference, the CFO calculated from different sets of signal samples may vary. To precisely decide the actual ZigBee channel being used, EmBee assumes that the CFO of each ZigBee channel as normal distributed, and use the ZigBee transmission history in the past to estimate the parameters of these distributions offline. As shown in Figure 5, these distributions are significantly different from each other. Decisions on ZigBee channels, then, are made based on maximum likelihood over these distributions.

V. Spectrum Reservation

EmBee's approach to spectrum reservation builds on the unique characteristics of modern Orthogonal Frequency Division Multiplexing (OFDM), which is the key PHY modulation technique for today's WiFi system. As shown in Figure 6, OFDM concurrently transmits data over 64 closely spaced subcarriers (48 data subcarriers, 4 pilot subcarriers and 12 null subcarriers with no power), each of which corresponds to a specific frequency band. Since these subcarriers are orthogonal to each other, EmBee is able to reserve the spectrum for an ongoing weak ZigBee transmission by avoiding the corresponding subcarriers from being used by WiFi, without affecting WiFi traffic in other subcarriers. We further reserve WiFi symbols partially to reduce the performance loss in WiFi without impairing weak Zigbee traffic.

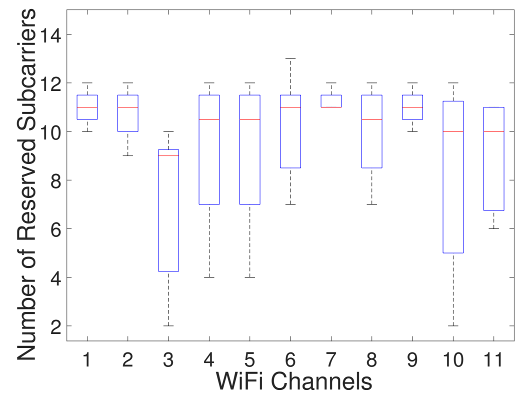


Figure 7: Number of reserved subcarriers for WiFi channels

A. Subcarrier Avoidance

In a 20 MHz WiFi channel, each OFDM subcarrier occupies $20M/64 = 312.5$ kHz spectrum. Therefore, in theory, a minimum number of $\lceil 2000/312.5 \rceil = 7$ subcarriers need to be reserved for a ZigBee channel with 2 MHz bandwidth. For example in Figure 6, subcarriers 6-13 need to be reserved to avoid collisions with ZigBee channel 16.

In practice, the required amount of spectrum being reserved may further increase because of sideband interference between consecutive subcarriers in the frequency domain. For example, WiFi traffic in subcarriers 5 and 14 may leak residue power into the spectrum of ZigBee channel 16 and hence result in ZigBee packet reception errors. We have experimentally investigated the practical impact of such sideband interference on spectrum reservation, by concurrently transmitting WiFi at 20 dB gain and ZigBee at 1 dBm Tx power, respectively. The amount of data subcarriers that need to be reserved for ZigBee transmission in different WiFi channels, as shown in Figure 7, is averaged at 9.75 and could be up to 13 in extreme cases. On the other hand, when the ZigBee channel overlaps with the 12 null subcarriers in WiFi, a smaller number of data subcarriers needs to be reserved. Note that, although different ZigBee channels may experience heterogeneous impact from WiFi's sideband interference and hence require different spectrum reservation as shown in Figure 7, such impact of sideband interference over one ZigBee channel remains constant in different network scenarios.

In addition, reserving some of the pilot subcarriers from being used by WiFi may affect channel prediction and equalization in the WiFi system. Our experiment results, as shown

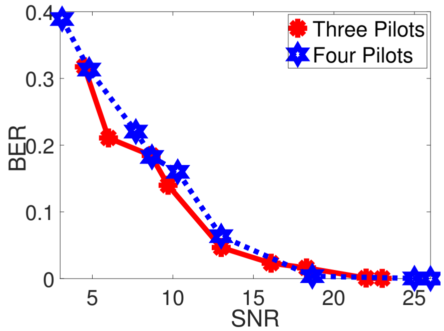


Figure 8: WiFi performance with 3 and 4 pilot subcarriers

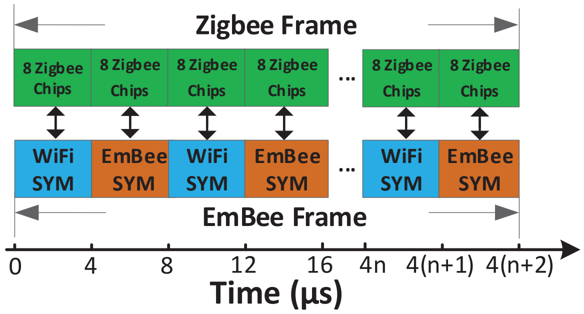


Figure 9: Minimizing WiFi throughput loss

in Figure 8, demonstrates that reserving one pilot subcarrier has very little impact to the WiFi performance. Since the maximum amount of spectrum reservation is 12 data subcarriers according to Figure 7 and the 4 pilot subcarriers (8, 22, 44 and 58) are at least separated by 14 subcarriers, so Embee will not reserve more than one pilot subcarrier in WiFi.

B. Minimizing WiFi Throughput Loss

According to Figure 7, the spectrum reservation approach in Section V-A could result in up to 25% throughput loss in WiFi. EmBee reduces such WiFi throughput loss by exploiting the built-in error correction capability in ZigBee: since every 4 data bits in ZigBee are transmitted as 32 chips, a ZigBee receiver maps every received sequence of 32 chips to the nearest 32-bit legitimate codeword measured by Hamming distance [2], and hence allows up to 20 transmission errors in every 32 chips [17].

As a result, EmBee does not need to reserve spectrum in every WiFi symbol. Instead, as shown in Figure 9, since every WiFi symbol lasts 4us and corresponds to 8 ZigBee chips, EmBee only needs to reserve spectrum in half of WiFi symbols in order to ensure 100% successful ZigBee packet reception, and correspondingly reduces the WiFi throughput loss to approximately 10%. Further reducing the amount of spectrum reservation improves the WiFi throughput at the cost of extra ZigBee packet reception errors, and we will experimentally investigate such tradeoff in Section VII-E.

C. Synchronization between WiFi Tx and Rx

The performance of EmBee may also be affected when the Embee Rx cannot precisely detect the reserved spectrum from Tx, because some of the reserved subcarriers may be corrupted by channel noise or distortion. As shown in Figure 10, if the

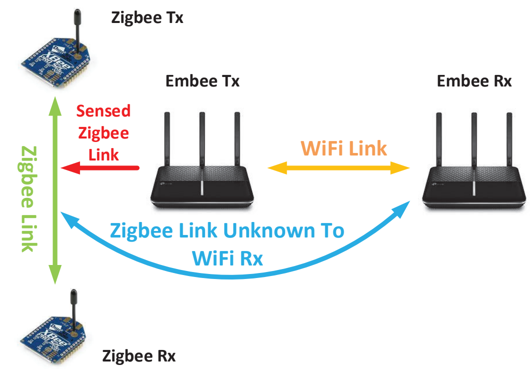


Figure 10: EmBee synchronization problem

Embee Rx has no precise knowledge of the reserved spectrum, it may produce extra WiFi packet reception errors. To address this synchronization problem, EmBee inserts one additional control byte into the beginning of payload in each WiFi data frame being sent, right after the preambles, to indicate the ZigBee channel for which WiFi spectrum is being reserved. As discussed in Section V-A, since the amount of spectrum reservation required by each ZigBee channel tends to remain constant over different network scenarios, the WiFi Rx is able to decide the right set of subcarriers to avoid according to the information about the ZigBee channel being used.

VI. Implementation

We have implemented EmBee over the WARP v3 software-defined radio testbeds, which is a FPGA-based wireless hardware platform and allows hardware-based implementation of a fully functional wireless transceiver over its onboard FPGA core. It hence allows realtime wireless networking and full compatibility with off-the-shelf wireless devices. Our implementation builds on the commodity 802.11g PHY that operates over the 2.4 GHz frequency band, and only requires minimum modification to the existing WiFi data flow by adding an extra subsystem for ZigBee detection and spectrum reservation. On the other hand, it does not require any modification on ZigBee devices.

The architecture of EmBee hardware implementation is shown in Figure 11, where the added subsystem is highlighted in color. It does not affect any existing module in the 802.11 PHY, but instead adds new subblocks after channel modulation in the Tx side and channel equalization in the Rx side.

At the Tx side, the strong WiFi transmitter will detect any ongoing weak ZigBee transmission before transmitting any data through WiFi channels. More specifically, after a wireless signal other than noise is being detected by channel sensing, it will calculate the CFO following the approach described in Section IV-A, by extracting one Zigbee sample from each ZigBee chip. Then, the calculated CFO will be compared with the pre-estimated parameters of different ZigBee channels to probabilistically decide which ZigBee channel is being used for transmission. Such decision will provide Zigbee channel information for the block of Spectrum Reservation, which avoids using the corresponding OFDM data subcarriers for WiFi transmission as described in Section V-A. Finally, before

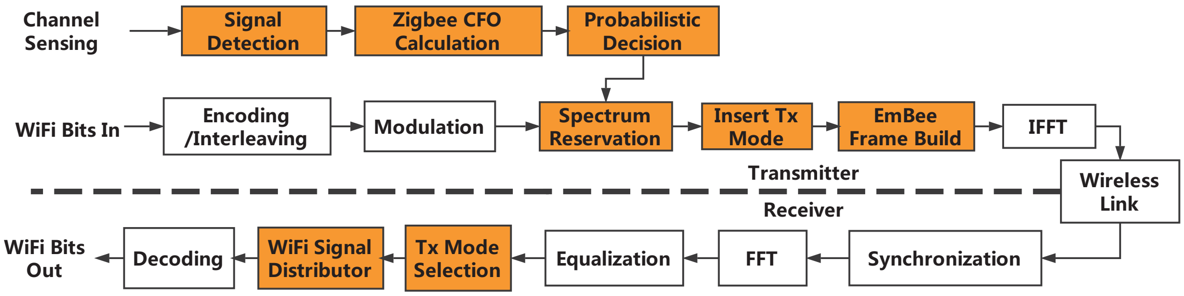


Figure 11: EmBee implementation

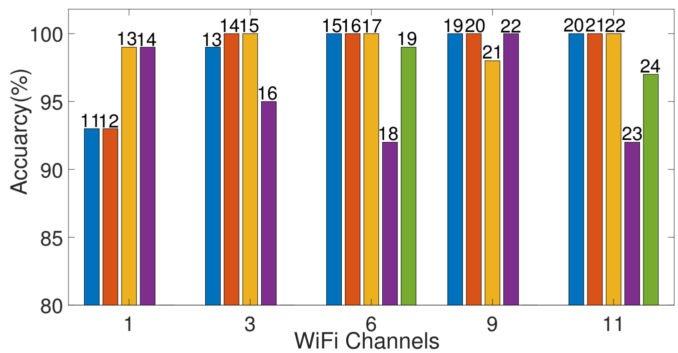


Figure 12: EmBee's accuracy of detecting ZigBee from WiFi. The numbers on top of bars indicate the index of ZigBee channel being detected.

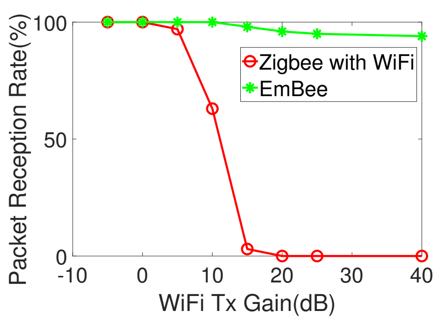


Figure 13: Zigbee packet reception rate with different levels of WiFi Tx gain

inverse FFT that translates data into time-domain samples, the Embee symbol building block inserts the modified WiFi symbols into a WiFi frame, as well as scheduling the spectrum reservation as described in Section V-B.

At the Rx side, after the received RF signal passes through channel equalization, a block of Rx Mode Selection will first decide the right spectrum being reserved by extracting the additional control byte being inserted by the WiFi Tx as described in Section V-C. Afterwards, a modified WiFi demodulator will be used to distribute the incoming signal onto the right set of OFDM subcarriers for correct data decoding.

VII. Performance Evaluation

Built on our implementation in Section VI, we evaluate the ZigBee performance achieved by EmBee under concurrent WiFi transmissions, as well as the corresponding WiFi performance loss caused by EmBee. We also compare the performance of EmBee with other cross-technology coexistence approaches. Evaluation results demonstrate that EmBee is able to always ensure weak ZigBee transmissions over a busy WiFi channel without incurring any extra ZigBee transmission delay, while resulting in minimal 10% of WiFi throughput loss.

A. Experiment Setup

We conduct our experiments in a 15m $\times$ 15m open space office with standard office furniture. Two WARP v3 SDR kits are used to operate EmBee over the 2.4 GHz WiFi frequency band, and constantly transmit WiFi packets with 1400-byte payloads at channels 1, 3, 6, 9 and 11. The WiFi Tx and Rx are placed 1.5 meters away from each other, and the WiFi Tx gain is adjusted from -5 dB to 30 dB to emulate different wireless link conditions. Both WARPs are connected to a Desktop PC with an Intel Xeon Gold 6144 CPU@3.5Ghz and 64GB of RAM for hardware configuration and data processing.

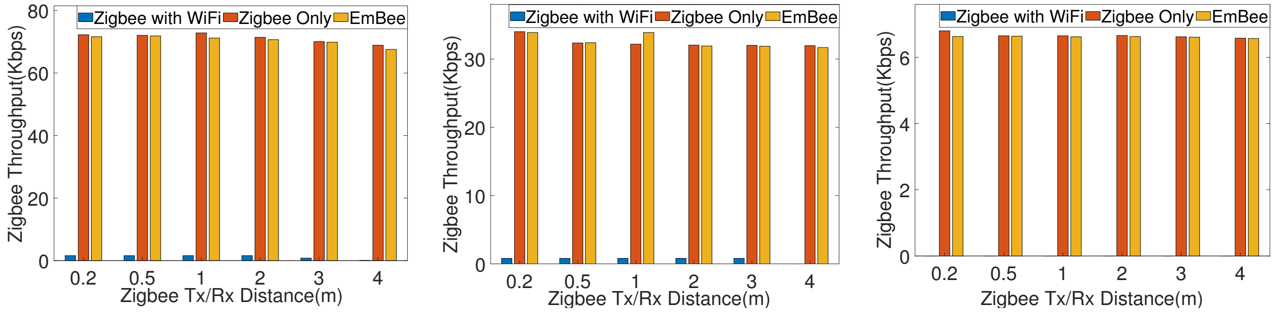
Meanwhile, two commodity Xbee S2C ZigBee transceivers are used to constantly transmit ZigBee packets with 84-byte payloads and 1 dBm Tx power. The communication distance ranges from 0.5 meter to 5 meters for different levels of signal strengths, and the ZigBee performance is being tested over all the ZigBee channels (11-25).

B. Accuracy of ZigBee Detection

We first evaluate the accuracy of EmBee's ZigBee detection approach proposed in Section IV. The channel offset of each ZigBee channel is averaged from 10k sets of ZigBee samples before being used for ZigBee channel detection, and each experiment is conducted 100 times for statistical convergence. The evaluation results over all the 5 WiFi channels being used are shown in Figure 12, which demonstrates that EmBee can achieve more than 93% accuracy when detecting ongoing ZigBee transmissions from a WiFi device, and such detection even reaches 100% accuracy in more than half of experiments over the most commonly used WiFi channel 6 and 11. Besides, the computational delay of such detection is averaged at 0.163ms, which is negligible to most of wireless applications.

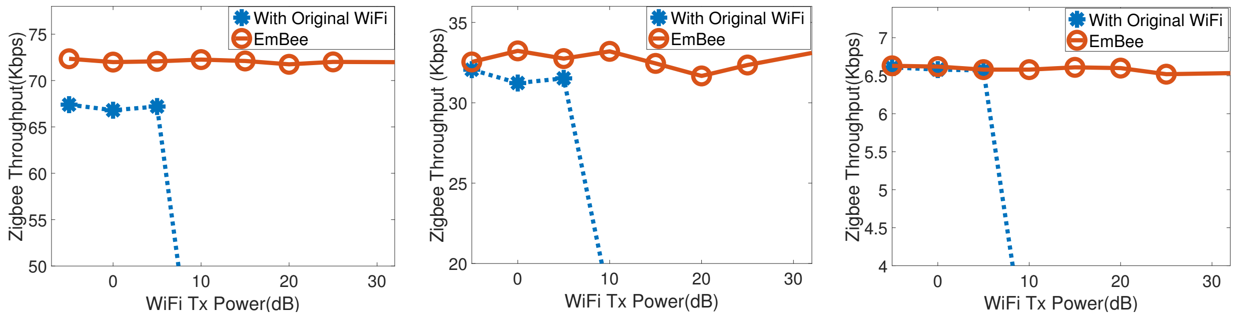
C. Zigbee Performance

In this section, we evaluate the ZigBee performance achieved by EmBee under concurrent WiFi traffic in the same spectrum. WiFi and ZigBee are operated in channel 6 and channel 16, respectively, and the ZigBee transceivers are placed 5cm away from the WiFi Tx to maximize the WiFi's



(a) 234000 Baud Rate (b) 57600 Baud Rate (c) 9600 Baud Rate

Figure 14: ZigBee throughput with different Zigbee communication distances



(a) 234000 Baud Rate (b) 57600 Baud Rate (c) 9600 Baud Rate

Figure 15: ZigBee throughput with different levels of WiFi Tx power

interference to ZigBee. According to Figure 7. EmBee reserves 10 data subcarriers from WiFi for ZigBee transmission.

1) Packet Reception rate: The Zigbee packet reception rates with different communication distances are listed in Table III. It shows that even when the WiFi channel is fully occupied by consistent traffic, EmBee is able to always ensure successful ZigBee packet reception. Furthermore, when the WiFi Tx gain is increased from -5 dB to 30 dB, Figure 13 shows that EmBee is able to always keep the ZigBee packet reception rate to be higher than 95%, while the original ZigBee transmissions will almost fail when the WiFi Tx gain is higher than 15 dB.

Table III: Zigbee Packet Reception Rate with different communication distances

Distance(m)	0.2	0.5	1	2	3	5
ZigBee w/ EmBee	1	0.99	1	1	1	1
ZigBee w/ WiFi@5dB Tx Gain	0.02	0.03	0.01	0.02	0.03	0.08
ZigBee w/ WiFi@20dB Tx Gain	0	0	0	0	0.01	0.05

2) ZigBee throughput: To evaluate the ZigBee throughput achieved by EmBee, we apply different baud rates to ZigBee transmissions, and also vary the ZigBee communication distance and WiFi Tx power that may affect the ZigBee throughput. Figure 14 shows that when the communication distance increases from 0.2 meter to 4 meters and hence the ZigBee Rx power significantly drops, the ZigBee throughput achieved by EmBee is always at the same level with that achieved by ZigBee in an idle channel without WiFi traffic. In contrast, original ZigBee transmissions in a busy WiFi channel will mostly fail due to collisions. Note that in all cases, the ZigBee throughput does not reach the theoretically maximum 250kbps due to hardware limits.

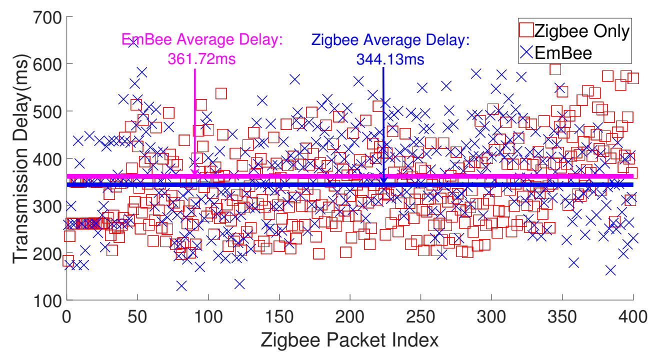


Figure 16: ZigBee transmission delay

Moreover, Figure 15 demonstrates that the ZigBee throughput achieved by EmBee also remains constant even if the WiFi Tx gain increases to 30 dB. In comparison, traditional ZigBee will fail when the WiFi Tx gain is higher than 10 dB.

3) Zigbee Delay: Figure 16 shows the delay of transmitting 400 ZigBee packets, and demonstrates that the ZigBee transmissions enabled by EmBee over a busy WiFi channel incurs only negligible extra delay. In particular, considering that the ZigBee retransmission time is usually higher than 200ms in cases of packet corruption, avoiding extra delay is particularly important for extremely weak wireless devices that are usually being used in time-sensitive wireless applications.

4) Performance Comparison: Finally, we compare EmBee with the existing cross-technology coexistence approaches, namely WISE [10] and Gsense [27], by varying the amount of WiFi traffic load, which is defined as the percentage of time that WiFi traffic occupies the channel. As shown in Figure 17, when the WiFi traffic load is below 50%, EmBee outperforms

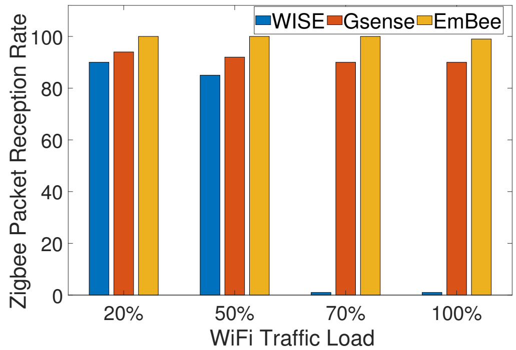


Figure 17: Comparison with existing coexistence approaches

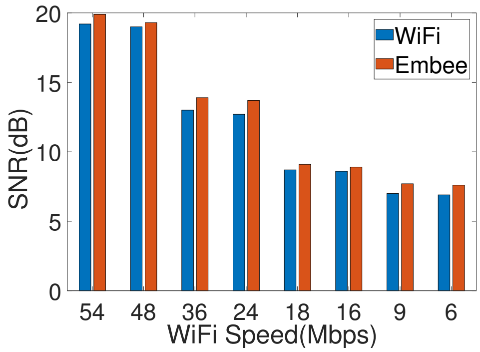


Figure 18: SNR requirement for 1% BER

WISE and Gsense by 15% and always ensures 100% ZigBee packet reception. However, if the WiFi traffic load further increases, both WISE and Gsense fail because of the lack of transmission opportunities. In contrast, the performance degradation of EmBee can be efficiently controlled within 1%, even with 100% WiFi traffic load. Considering that EmBee does not introduce any extra ZigBee transmission latency, it significantly outperforms these existing approaches and is particularly suitable for extremely weak wireless devices with limited transmission opportunities.

D. WiFi Performance

In this section, we evaluate EmBee's impact on the strong WiFi performance. Our experiments transmit 1000 WiFi packets with a 1400-byte payload, during which ZigBee is enabled by EmBee to continuously transmit. We first investigate the minimum WiFi channel SNR that is required to achieve 1% Bit Error Rate (BER) for different WiFi data rates. As shown in Figure 18, for all 802.11g WiFi data rates ranging from 6 Mbps to 54 Mbps, EmBee requires only 1 dB additional SNR to reach the 1% BER. We then evaluate the effectiveness of our proposed technique in Section V on reducing the WiFi throughput loss. As shown in Table IV, for all the designated WiFi data rates, EmBee effectively controls the WiFi throughput loss within 12%. In particular, such throughput loss gradually decreases when the WiFi data rate drops. EmBee, hence, is more suitable to be applied to severe application scenarios with poor wireless signal quality.

E. Tradeoff Between ZigBee and WiFi Performance

As discussed in Section V, the error correction capability of ZigBee allows EmBee to further balance between ZigBee

Table IV: WiFi throughput loss caused by EmBee

Data Rate (Mbps)	54	48	36	24	18	12	9	6
Original WiFi (Mbps)	25.82	23.62	17.53	13.93	10.88	8.13	6.30	4.61
WiFi w/ EmBee (Mbps)	22.57	20.88	15.56	12.41	9.68	7.24	5.61	4.12
Throughput loss (%)	12.60	11.60	11.30	10.90	11.10	10.90	11.00	10.70

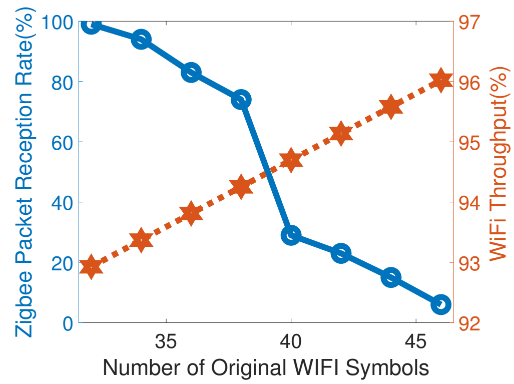


Figure 19: Tradeoff between ZigBee and WiFi performance

and WiFi performance, by adjusting the amount of spectrum being reserved for ZigBee transmissions. To investigate such tradeoff, we divide the input WiFi symbols into groups of 64 symbols, and then vary the number of WiFi symbols in each group. As shown in Figure 19, when such number increases from 32 to 46, the theoretical WiFi throughput loss could be effectively reduced to 4%, at the cost of reducing the practical ZigBee packet reception rate. In general, EmBee is able to control the WiFi throughput loss within 10% with minor degradation of ZigBee performance. On the other hand, in extreme cases which always require reliable ZigBee data delivery, EmBee can also satisfy this requirement by increasing the WiFi throughput loss.

VIII. Related Work

Cross-technology coexistence has been well studied by the research literature. Some researchers added a cooperative busy tone into the 802.15.4 MAC layer, allowing the low-power Zigbee transmission being detected by strong WiFi to reduce packet collision [26]. Other research efforts aim to modify the ZigBee MAC. Gsense [27] customized the Zigbee preamble to reduce the collision with WiFi, but requires a weak ZigBee device to actively send customized signal onto the air and hence reduces the ZigBee throughput. WISE [10] proposed a new Zigbee MAC design to increase the Zigbee throughput in the existence of concurrent WiFi transmissions, by predicting the white space between WiFi frames and then utilizing such white space for ZigBee transmissions. BuzzBuzz [17] customizes the Zigbee packet with an extra header to increase the Zigbee packet reception rate, by finding whether the extra header is corrupted for timely retransmission. However, most of these schemes require hardware modifications over ZigBee devices, and introduce extra delay and link quality degradation to ZigBee transmissions [3]. Hence, they cannot be applied to extremely weak wireless devices with limited transmission opportunities or local resources.

Other researchers strive to explore PHY-layer solutions to cross-technology coexistence. ZIMO separates WiFi and Zigbee signals into different data streams by using MIMO and interference cancellation [25], so as to avoid collisions among them. However, it requires completely new ZigBee transceiver design with extra hardware and signal procesing, which is unaffordable by weak wireless devices.

Cross-technology communication is another way of achieving cross-technology coexistence, by enabling explicit coordination between wireless technologies to avoid transmission collisions. FreeBee [11] enables cross-technology communication by adding a modulated beacon that can be demodulated from heterogeneous devices. WEBee [16] further extended the beacon by using software to emulate Zigbee packets from WiFi devices, and B2W2 [6] enables concurrent communication among multiple WiFi and Bluetooth devices by embedding a Bluetooth message into its overlapped WiFi subcarriers. However, such explicit coordination inevitably incur additional delay to ZigBee transmissions. Chiron [12] avoids such additional delay by allowing WiFi and ZigBee data streams to concurrently transmit in the same spectrum, but only applies to data traffic to/from a customized data gateway. It hence may also increase the data transmission delay between weak wireless devices, when relaying data between these devices via the gateway. Instead, EmBee completely avoids any extra transmission delay over weak ZigBee devices, without requiring any hardware modification over these devices.

The design of EmBee is also inspired by existing work on wireless side channel, which could be created from a portion of WiFi subcarriers and then used for data transmissions of other wireless technologies. Existing approaches to wireless side channel design, however, are limited to transmitting data generated by the sole wireless device [18], [24]. EmBee extends these designs by allowing wireless technology detection and spectrum reservation across multiple wireless devices operating with different wireless technologies.

IX. Conclusion

In this paper, we present EmBee, a new approach to enable cross-technology coexistence for extremely weak wireless devices that have limited transmission opportunities and amounts of local resources. The key idea of EmBee is to allow the strong wireless devices to be aware of the ongoing wireless transmissions from weak devices, so as to reserve spectrum for such transmissions accordingly. EmBee realizes this idea by utilizing the unique characteristics of modern OFDM modulation in WiFi, so as to support concurrent ZigBee transmissions over the same spectrum of a busy WiFi channel without incurring any extra delay or throughput loss to ZigBee. At the same time, EmBee effectively controls the WiFi throughput loss due to such spectrum reservation to 10%, which is negligible to most wireless applications.

References

[1] Coexistence analysis of ieee std 802.15.4 with other ieee standards and proposed standards. *IEEE 802.15 Working Group*, 2010.

[2] Part 10: O-QPSK PHY. *IEEE Standard 802.15.4-2011*, Sep 2011.

[3] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: A survey. *ACM Transactions on Sensor Networks (TOSN)*, 2012.

[4] B. Bloessl, M. Segata, C. Sommer, and F. Dressler. An ieee 802.11 a/g/p ofdm receiver for gnu radio. In *Proceedings of the second workshop on Software radio implementation forum*, pages 9–16. ACM, 2013.

[5] K. Chebrolu and A. Dhekne. ESense: communication through energy sensing. In *Proceedings of ACM MobiCom*. ACM, 2009.

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

[7] M. Communications. 802.11 reference design for warp v3. In *warpproject.org/trac/wiki/802.11*, 2013.

[8] R. Friedman, A. Kogan, and Y. Krivolapov. On power and throughput tradeoffs of wifi and bluetooth in smartphones. *IEEE Transactions on Mobile Computing*, 12(7):1363–1376, 2013.

[9] I. Howitt and J. A. Gutierrez. Ieee 802.15. 4 low rate-wireless personal area network coexistence issues. In *Proceedings of IEEE WCNC*, 2003.

[10] J. Huang, G. Xing, G. Zhou, and R. Zhou. Beyond co-existence: Exploiting wifi white space for zigbee performance assurance. In *In Proceedings of ICNP*, 2010.

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

[12] Y. Li, Z. Chi, X. Liu, and T. Zhu. Chiron: Concurrent high throughput communication for iot devices. In *Proceedings of ACM MobiSys*. ACM, 2018.

[13] Y. Li and W. Gao. Interconnecting heterogeneous devices in the personal mobile cloud. In *Proceedings of IEEE INFOCOM*, 2017.

[14] Y. Li and W. Gao. Minimizing context migration in mobile code offload. *IEEE Transactions on Mobile Computing*, 16(4):1005–1018, 2017.

[15] Y. Li and W. Gao. MUVR: Supporting multi-user mobile virtual reality with resource constrained edge cloud. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018.

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

[17] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving wi-fi interference in low power zigbee networks. In *In Proceedings of ACM SenSys*, 2010.

[18] H. Lu and W. Gao. Supporting real-time wireless traffic through a high-throughput side channel. In *In Proceedings of ACM MobiHoc*, 2016.

[19] H. Lu and W. Gao. Continuous wireless link rates for internet of things. In *Proceedings of the ACM/IEEE Int'l Conference on Information Processing in Sensor Networks*, 2018.

[20] S. Pollin, M. Ergen, M. Timmers, A. Dejonghe, L. Van der Perre, F. Catthoor, I. Moerman, and A. Bahai. Distributed cognitive coexistence of 802.15. 4 with 802.11. In *Crowncom*, pages 1–5, 2006.

[21] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung. A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. *IEEE Wireless Communications*, 20(6):91–98, 2013.

[22] T. Wild, F. Schaich, and Y. Chen. 5G air interface design based on universal filtered (UF-)OFDM. In *IEEE DSP*, 2014.

[23] C. Won, J. Youn, H. Ali, H. Sharif, and J. Deogun. Adaptive radio channel allocation for supporting coexistence of 802.15. 4 and 802.11 b. In *IEEE Vehicular Technology Conference*. IEEE; 1999, 2005.

[24] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. Ni. Side channel: Bits over interference. In *In Proceedings of ACM Mobicom*, 2010.

[25] Y. Yubo, Y. Panlong, L. Xiangyang, T. Yue, Z. Lan, and Y. Lizhao. Zimo: Building cross-technology mimo to harmonize zigbee smog with wifi flash without intervention. In *In Proceedings of ACM MobiCom*, 2013.

[26] X. Zhang and K. Shin. Enabling coexistence of heterogeneous wireless systems: Case for zigbee and wifi. In *In Proceedings of ACM MobiHoc*, 2011.

[27] X. Zhang and K. G. Shin. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *IEEE INFOCOM*, 2013.

[28] J. Zhu, A. Waltho, X. Yang, and X. Guo. Multi-radio coexistence: Challenges and opportunities. In *Proceedings of IEEE ICCCN*, 2007.