WEBee: Physical-Layer Cross-Technology Communication via Emulation

Zhijun Li*
University of Minnesota
Minneapolis, MN
lizhijun.hit@gmail.com

Tian He†
University of Minnesota
Minneapolis, MN
tianhe@umn.edu

ABSTRACT

Recent advances in Cross-Technology Communication (CTC) have improved efficient coexistence and cooperation among heterogeneous wireless devices (e.g., WiFi, ZigBee, and Bluetooth) operating in the same ISM band. However, until now the effectiveness of existing CTCs, which rely on packet-level modulation, is limited due to their low throughput (e.g., tens of bps). Our work, named WEBee, opens a promising direction for high-throughput CTC via physical-level emulation. WEBee uses a high-speed wireless radio (e.g., WiFi OFDM) to emulate the desired signals of a low-speed radio (e.g., ZigBee). Our unique emulation technique manipulates only the payload of WiFi packets, requiring neither hardware nor firmware changes in commodity technologies – a feature allowing zero-cost fast deployment on existing WiFi infrastructure. We designed and implemented WEBee with commodity devices (Atheros AR2425 WiFi card and MicaZ CC2420) and the USRP-N210 platform (for PHY layer evaluation). Our comprehensive evaluation reveals that WEBee can achieve a more than 99% reliable parallel CTC between WiFi and ZigBee with 126 Kbps in noisy environments, a throughput about 16,000x faster than current state-of-the-art CTCs.

CCS CONCEPTS

• **Networks** → Wireless Networks;

KEYWORDS

Cross Technology Communication; Signal Emulation; WiFi OFDM; ZigBee OQPSK, DSSS

1 INTRODUCTION

According to Gartner [15], the number of Internet of Things (IoT) devices connected wirelessly will reach 20 billion by 2020, leading to the intense coexistence of wireless technologies.

For spectrum efficiency under dense deployment, many of today's wireless technologies are designed to share the unlicensed spectrum (e.g., ISM bands), including such popular technologies as WiFi, Bluetooth, and ZigBee. Despite the common belief that coexistence of wireless technologies leads harmful interference, it in fact offers new opportunities for those technologies to collaborate. Recent research shows that cross-technology communication (CTC), defined as direct communication (i.e., message/data exchange) among heterogeneous wireless devices, can bring about many benefits. For example, ZiFi [50] can significantly reduce the standby energy of WiFi devices when a low-power ZigBee radio is used to wake up the WiFi NIC whenever it detects the existence of WiFi APs, and FreeBee [23] can reduce this by 58% more if the open/private setting of a WiFi AP is conveyed through CTC. Furthermore, CTC will provide more efficient channel coordination by exchanging global RTS/CTS and TDMA messages *explicitly* among coexisting heterogeneous devices, in replace of *implicit* clear channel assessment (CCA).

Traditionally, bridging wireless technologies was achieved indirectly through multi-radio gateways, which introduce extra hardware cost, deployment complexity, and doubling the traffic into and out from these gateways. To avoid these drawbacks, a few pioneering works [5, 9, 20, 23] support CTC among heterogeneous wireless devices despite their incompatible physical layer modulation. We note that existing CTCs use packet-level modulation – embedding symbols using the packet length [5], timing [23], and sequence patterns [20, 43]. For example in [43], a sender transmits a sequence of long and short Wi-Fi packets to construct Morse codes, which can be decoded by receivers through RSSI energy sensing. With packet-level modulation, a packet can express at most a few bits in CTC [5], as opposed to thousands of bits if the packet were used for intra-technology communication [18]. Furthermore, early solutions [9, 49] require sending dummy packets for CTC, which wastefully occupy the channel, leading to further inefficiency.

In this paper, we introduce a new direction for high-throughput CTC, a physical-level emulation technique named WEBee (for WiFi Emulated ZigBee). In a nut-shell, WEBee chooses the payload of a WiFi frame in such a way as a portion of this WiFi frame is recognized by commodity ZigBee devices transparently as a legitimate ZigBee frame. Since we modify neither the firmware nor hardware of the WiFi transmitter, this WiFi frame can also be legitimately received by WiFi receivers. We note that the pioneering work [19] produces RFID AM signals using Wi-Fi devices, which inspires us to create physical-level cross-technology communication that is dual standard compliance (i.e., ZigBee and WiFi).

*Zhijun Li is a visiting professor at the University of Minnesota and officially affiliated with Harbin Institute of Technology, China.
†Tian He is the corresponding author.

MobiCom '17, October 16–20, 2017, Snowbird, UT, USA

ACM ISBN 978-1-4503-4916-1/17/10…$15.00
https://doi.org/10.1145/3117811.3117816

Table 1: Motivation of the WEBee

	Cost	Spectrum Efficiency	Throughput	Parallel CTCs
Gateway	Median	Median	High	Not Support
FreeBee[23]	Low	Median	Low	Not Support
Esense[5]	Low	Low	Low	Not Support
B^2W^2[9]	Low	Low	Low	Not Support
WEBee	**Low**	**High**	**High**	**Support**

To support CTC, a WiFi transmitter needs to emulate the ZigBee time-domain waveform closely enough so that it can pass ZigBee preamble detection and allow successful OQPSK demodulation. With such signal emulation, WEBee conceptually can achieve a 250Kbps data rate from WiFi to ZigBee, the ceiling speed of standard ZigBee communications. In practice, after accommodating emulation errors, a WEBee WiFi transmitter achieves a 99.9% symbol reception ratio at 63Kbps data rate, which is more than 8,000 times faster than the reliable free-channel rate (7.5bps@99%) reported by FreeBee [23]. Moreover, since WiFi occupies a much wider bandwidth (20MHz) than ZigBee (2MHz), WEBee can successfully emulate two ZigBee frames under different frequencies within a single WiFi frame, resulting in a 16,000x throughput and a higher spectrum efficiency.

This paper presents the first emulation-based CTC design from WiFi to ZigBee specifically. However, the features we provide and the challenges we address in this emulation-based design are indeed generic and applicable to a whole set of future physical-layer CTCs. Specifically, the major contributions of WEBee are as follows:

- We design WEBee, a CTC technique that emulates ZigBee frames with the payload of WiFi frames. Without modifying the firmware or hardware of both WiFi and ZigBee devices, WEBee is a transparent design that be can easily deployed in existing WiFi infrastructure with broad applicability.
- To extend the range, we address a few unique challenges, which include (i) optimized ZigBee signal emulation using the WiFi OFDM modulation, (ii) reverse WiFi channel coding mapping, (iii) pilot/null subcarrier avoidance, (iv) parallel CTC, and (v) link-level reliability. These techniques provide general guidances for range extension of emulation-based CTC designs.
- We implement and evaluate WEBee on commodity devices (Atheros AR2425 WiFi NIC and MicaZ CC2420) and the USRP-N210 platform (for PHY-layer evaluation). Our extensive evaluation demonstrates that WEBee can achieve a fast, robust, and parallel CTC performance under a full range of wireless configurations, including stationary, mobile, long-distance, and duty-cycled settings. In all these settings, the frame reception ratio (FRR) reaches above 99% with six retransmissions, and throughput is orders of magnitude larger than existing CTC solutions [5, 9, 23, 43]. We have also embedded WEBee into Nexus 5 smart phones with BCM4330 WiFi chips to control Zigbee Smart bulb directly, indicating WEBee is applicable in today's mainstream devices as well [27].

2 MOTIVATION

Spectrum efficiency calls for sharing within the unlicensed bands. Between 2014-2016, the FCC [12–14] opened the spectrum in 600MHz, 5GHz, and 7GHz, bringing in new ISM standards, including LTE-U (by T-Mobile and Verizon Wireless [33]) and 802.11ah (i.e., WiFi HaLow [41]). Such coexisting environments necessitate explicit information exchange among heterogeneous wireless devices to support channel coordination and cross-technology collaboration. Information can be exchanged through either multi-radio gateways or cross-technology communication (CTC). In this section, we examine the limitations of existing solutions and summarize the benefits of WEBee in Table 1.

Limitation of Gateways: Clearly, multi-radio gateways can serve as translators among heterogeneous wireless devices with incompatible physical layers. However, this solution is inherently limited in several aspects. First, gateways incur extra hardware costs and deployment complexity, in which the density and physical locations of the gateways noticeably impact network performance. Second, gateway-based approaches induce significant traffic overhead flowing in and out from the gateway, which further intensifies interference, especially in multi-hop scenarios. Third, gateways have to be deployed in advance in the situations where CTCs are required, making it difficult to support mobile and ad hoc scenarios.

Limitation of Packet-Level CTC: The inherent limitation of the packet-level CTCs is their throughput. Usually, the duration of a wireless packet is in the range of milliseconds. Embedding CTC symbols with sparse packet-level information (i.e., the packet Tx timing in FreeBee [23], the packet RSSI duration in Esense [5]) limits the bandwidth. For example, in ZigBee devices, the packet-level RSSI information is sampled at only 31.25KHz, while the phase-shift is obtained at 4MHz by the ZigBee PHY for (de)modulation. In other words, the throughput of packet-level CTCs is inherently bounded by a low sampling rate (e.g., KHz) in contrast to hundreds of Kbps for native ZigBee communication.

Another limitation of the packet-level CTCs is that they fail to utilize bandwidth fully. A single WiFi transmission occupies 20-40MHz channels, while ZigBee receivers only obtain packet-level information within a 2MHz-wide ZigBee channel. We note that this is an inherent limitation of packet-level CTCs because the packet-level information (e.g., packet Tx timing) perceived by ZigBee at different channels is the same, making parallel symbol transmission infeasible at the packet level.

Advantage of Physical-Level CTC: As summarized in Table 1, WEBee resolves the limitation of gateways by offering direct connectivity. For example, it allows a WiFi AP to control all IoT devices equipped with low-power ZigBee radios in a smart home without a gateway. Instead of depending on sparse packet-level information,

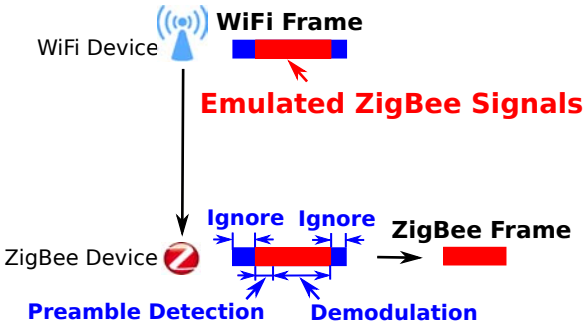


Figure 1: The Architecture of the WEBee

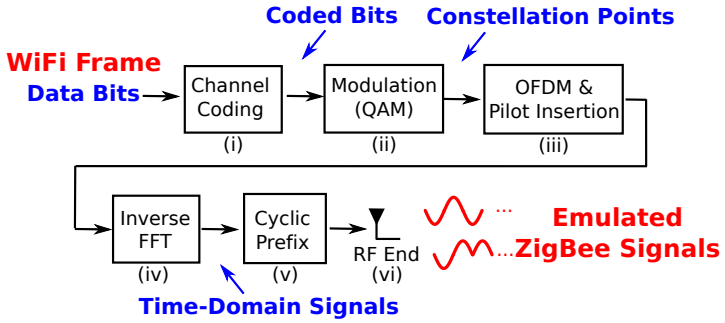


Figure 2: How WiFi Transmitter Works

WEBee achieves a throughput close to that of the ZigBee standard by emulating time-domain ZigBee signals. Parallel CTCs become feasible by emulating multiple time-domain ZigBee signals at different frequency bands, further boosting the total throughput.

3 OVERVIEW OF THE WEBEE DESIGN

Figure 1 illustrates how WEBee works: a WiFi device transmits a frame with a *selected* payload that can be recognized by a ZigBee receiver as a *legitimate* ZigBee frame. More specifically, WEBee constructs the payload of a WiFi frame *elaborately* so that the RF waveform of the payload resembles that of ZigBee signals. When such a WiFi frame is transmitted into the air through the front end of the WiFi RF, the WiFi header, preamble, and trailer are ignored by ZigBee receivers as noise, while the payload will successfully pass the ZigBee preamble detection and the emulated ZigBee frame will be then demodulated at the ZigBee receiver. We note that WEBee is indeed a *transparent* design, in the sense that a ZigBee receiver cannot differentiate whether the sender is a ZigBee or a WiFi device. Moreover, time-domain waveforms of multiple ZigBee frames can be modulated into one WiFi frame because of the wider band of WiFi transmissions. Multiple ZigBee receivers working on different channels can detect and demodulate different emulated ZigBee frames simultaneously and independently.

ZigBee接收方并不能分辨出发送方是WiFi还是ZigBee

3.1 Background

To explain WEBee, it is necessary to first introduce how WiFi transmitters and ZigBee receivers work. Although our description is specific, our approach is generically applicable to future emulation-based CTCs.

3.1.1 How a WiFi Transmitter Works. Figure 2 shows how a WiFi transmitter (802.11g/n) works, from step (i) to step (vi). (i) The channeling coding module encodes the data bits in a WiFi frame into redundant coded bits for robustness. (ii) Then these coded bits are mapped into a set of constellation points based on selected modulation schemes, typically Quadrature Amplitude Modulation (QAM). (iii) By using Orthogonal Frequency Division Multiplexing (OFDM), these constellation points are modulated into 48 data subcarriers, while additional pseudo-random pilot symbols are modulated into pilot subcarriers for channel estimation at the WiFi receivers. (iv) After that, Inverse Fast Fourier Transform (IFFT) combines all these subcarriers and turns them into a time-domain signal. (v) The WiFi time-domain signal is then processed by the cyclic prefix module, which prefixes a WiFi symbol with a repetition of the end. By cyclic prefixing, a guard 0.8us interval is formed to eliminate inter-symbol interference. (vi) Finally, a whole WiFi symbol with a 4μs duration is generated at the WiFi sender and is sent by the WiFi RF radio.

The objective of WEBee is to create time-domain waveforms that can be recognized by ZigBee receivers. To emulate ZigBee waveforms, WEBee proceeds with a reverse direction from step (vi) to step (i) as shown in Figure 2. To transmit the desired time-domain ZigBee RF signals at step (vi) from the commodity WiFi radios, at step (iii) WEBee needs to choose corresponding frequency components, which are mapped from the set of constellation points (complex numbers) selected at step (ii). These constellation points are controlled by the source bits of the WiFi payload.

In other words, source bits in a *selected* WiFi payload determine the QAM constellation points after the WiFi modulator. When a specific combination of constellation points are fed into the IFFT, the desired time-domain signals are emulated and transmitted from the commodity WiFi radios. With such signal emulation, the commodity ZigBee radios can demodulate and decode the "frame" sent from the WiFi to achieve CTC. It is noted that WEBee does not change the WiFi modulator, and therefore QAM-based emulation contains an intrinsic error because only a limited number of discrete QAM constellation points are available for emulation.

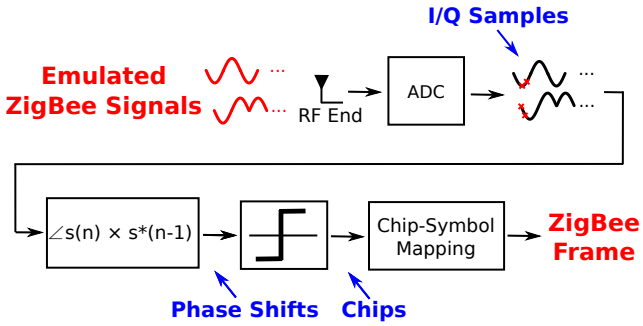


Figure 3: How ZigBee Receiver Works

3.1.2 How a ZigBee Receiver Works. WEBee sends WiFi frames to commodity ZigBee devices. The physical layer of ZigBee receivers is shown in Figure 3. Because of sharing the same ISM wireless band (e.g., 2.4GHz), the waveform emulated by WiFi can

be sampled by ZigBee devices. To receive a frame, a ZigBee RX Radio down-converts the received WiFi passband waveforms to baseband and digitalizes them into in-phase and quadrature (I/Q) samples using ADC.

The modulation of ZigBee is Offset Quadrature Phase-Shift Keying (OQPSK), in which the phase shifts between the consecutive I/Q samples are used to demodulate the ZigBee symbols. Specifically, the phase shift between two consecutive complex samples $s(n)$ and $s(n-1)$ is computed by $arctan(s(n) \times s^*(n-1))$, where $s^*(n-1)$ is the conjugate of $s(n-1)$. With thresholding, ZigBee outputs the chip value "1" if the phase shift is bigger than 0° and otherwise outputs the chip value "0". After collecting 32 chips, ZigBee maps these chips into four bits (i.e., a ZigBee symbol in frame), according to the predefined symbol-to-chip spreading relationship in the direct sequence spread spectrum (DSSS) process.

3.1.3 Why Emulation is Feasible. In ZigBee (i.e., IEEE 802.15.4), direct sequence spread spectrum (DSSS) is used to improve protection against interference and noise by multiplying original bits with a pseudo random noise spreading code. Specifically, a ZigBee symbol (i.e., 4-bits) are mapped into a 32-chip sequence. In DSSS, a correlation threshold is defined to control the maximum Hamming distance between the received 32-chip sequence and the predefined chip sequence that the receiver can tolerate. A none-zero threshold means that ZigBee receivers inherently can tolerate a certain number of errors in chip sequences. Normally, 12 is the default threshold, i.e., 12 chip errors can be recovered by the ZigBee DSSS technique. In some scenarios, this threshold is set more loosely to adapt to environments with high interference and noise. For example, in [28], the default of such a threshold is set to 20 for ZigBee frames to survive WiFi interference. As mentioned early, QAM-based emulation could introduce errors in chip sequences. However, such errors are mediated by DSSS, and the emulated ZigBee symbols can be decoded by ZigBee successfully with a high probability.

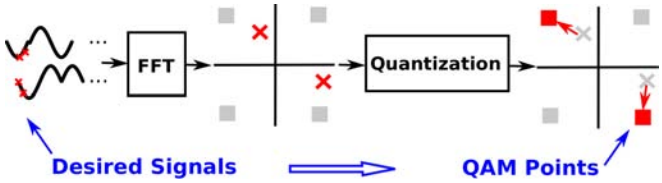


Figure 4: The Basic Process of QAM Emulation.

3.2 QAM Emulation

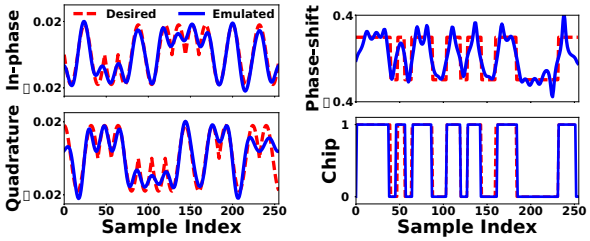
QAM emulation is the core of the WEBee design. Figure 4 shows that the QAM selection is done in the reverse direction, where the desired ZigBee time-domain signals are fed into the FFT to select corresponding QAM constellation points. The challenging issue of QAM emulation is that because the QAM points in WiFi OFDM are predefined and discrete, the frequency components of the desired time-domain signals might not match the discrete QAM points perfectly, as shown in Figure 4, leading to intrinsic QAM quantization errors.

In Fourier transform analyses, the Parseval's theorem [42]states an equation of energy in the time-domain and frequency-domain. Combined with the linear property of Fourier transform, we have the following equation for the errors introduced by frequency-domain quantization, i.e.,

$$\int_{t=-T/2}^{T/2} |u(t) - v(t)|^2 dt = T \sum_k |U[k] - V[k]|^2, \qquad (1)$$

where $u(t)$ is the desired time domain signal, $v(t)$ is the time domain signal after QAM quantization, and $U[k]$ and $V[k]$ are corresponding DFTs.

The difference-energy equation shows that minimizing the signal distortion in the time-domain under energy metric is equivalent to minimizing the total deviation of frequency components after QAM quantization. Therefore QAM emulation is essentially an optimization process to choose the closest k QAM points in term of total Euclidean distance to each of k FFT points of desired signals as shown in Figure 4, which can be done easily in $O(k)$.



(a) Emulated time-domain signal **(b) Phase shifts of emulated signal**

Figure 5: Comparison between WiFi QAM Emulated Signal and the Desired ZigBee Signal

As a proof of concept example, an emulated ZigBee symbol "5" is emulated according to the process shown in Figure 4, where the 64-QAM constellation points in IEEE 802.11g standard are utilized for emulation. After the IFFT, the emulated time-domain signal is shown in Figure 5a, where the real/imaginary parts of the emulated time-domain signal are compared with the desired In-phase/Quadrature (or I/Q) signal of the ZigBee symbol. It is easy to see that the desired ZigBee signals are approximated well by the emulated WiFi signal, with some tolerable distortion.

Furthermore, the phase shifts of the emulated signal (and corresponding chip values) are calculated and shown in Figure 5b. The results prove that the emulated signal can be decoded by ZigBee successfully.

3.3 Emulation under Hardware Constraints

For the sake of clarity, the QAM emulation illustrated in Figures 4 and 11 are introduced without mentioning hardware constraints. Clearly, to support cross-technology communication from WiFi to ZigBee in commodity devices, WEBee needs to handle a few challenging issues. Figure 6 shows the detailed transmission path of WiFi OFDM. In addition to QAM emulation, we need to support two additional types of emulation: (i) channel coding emulation and (ii) post-QAM emulation. The challenges related to each of these emulations are discussed in section 4 and section 5, respectively.

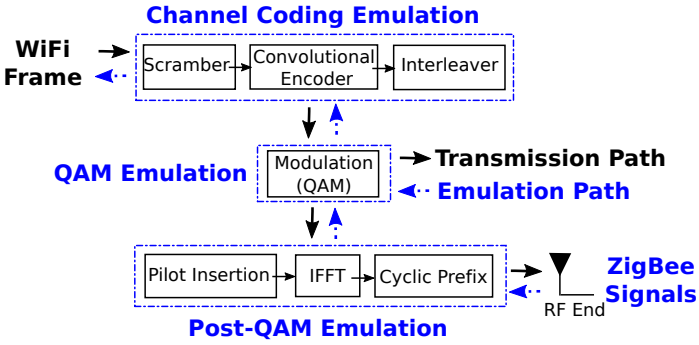


Figure 6: Complete WEBee Emulation Procedure.

4 CHANNEL CODING EMULATION

It is known that channel coding (convolutional encoding) makes WiFi OFDM communications more resilient to noise and interference by introducing redundancy (extra bits). But this imposes a challenge for emulation from the reverse direction because we can map source bits in the WiFi payload into only a *constrained* set of coded bits to activate QAM points for OFDM subcarrier modulation.

More specifically, in WiFi 64-QAM, six coded-bits are used to map one QAM point. Given 48 QAM points, a total of 288 (= 48×6) coded bits are used. Given a channel coding rate of 3/4, we use the 216 (= $288 \times \frac{3}{4}$) bits in the WiFi payload to generate the 288 coded bits.

If matrices in Galois field GF(2) are used to define the relationship between the source bits X and the coded bits Y, it is easy to model the convolutional encoding as a matrix M satisfying the following equation:

$$M \times_{GF(2)} X = Y \tag{2}$$

For any Y, finding the X satisfying the equation (2) is equivalent to finding the inverse of M under GF(2). If M is full row-rank and square, the inverse of M can be calculated easily by Gaussian Elimination.

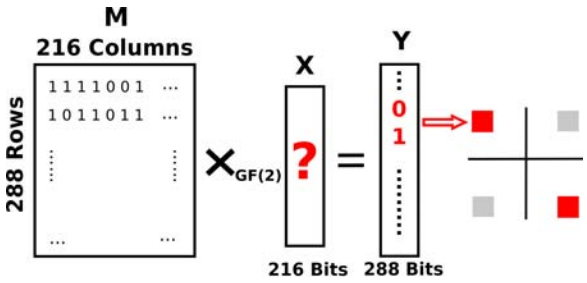


Figure 7: Invertible QAM Points under Constraints of WiFi Channel Coding

However, as shown in Figure 7, to map 216 source bits in X to 288 coded bits in Y, convolutional encoding uses a 288-by-216 matrix M. Since M is not full row-rank (row:288 > column:216), the matrix equation is an overdetermined system (i.e., more equations than unknowns). In other words, WEBee cannot emulate an arbitrary combination of 48 QAM points using 216 source bits.

But luckily, we note, ZigBee signals occupy only a 2MHz band, covering 7 WiFi subcarriers with a 312.5KHz bandwidth each. To emulate ZigBee signals, WEBee needs to control only 7 WiFi QAM points (14 QAM points in the parallel case) instead of all 48 QAM points. According to Equation 2, WEBee needs to control only 84 bits (14 ×6 bits) of Y by manipulating the X. Let Y' be the 84-bits subvector of Y and M' be the corresponding submatrix, we have,

$$M' \times_{GF(2)} X = Y'. \tag{3}$$

Given M' is now a full row-rank matrix (row:84 < column:216), for arbitrary coded bits Y, the matrix equation (3) is an underdetermined system with multiple solutions of X. In other words, WEBee can emulate an arbitrary combination of 14 QAM points with 216 source bits in multiple ways.

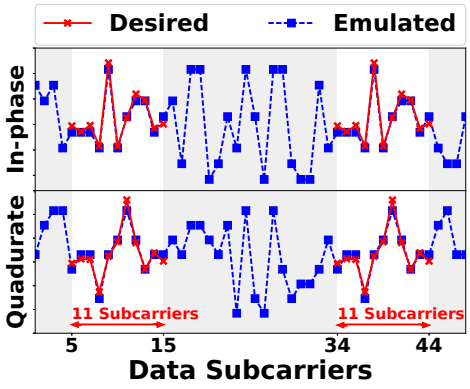


Figure 8: An Example of QAM Approximation by Controlling a Subset of Subcarriers

Our experiment in Figure 8 indicates that WEBee emulates ZigBee closely in two subregions of 11 subcarriers (7 data + 4 guard subcarriers) using 216 source bits. These two subregions overlapped with 2MHz ZigBee channels with 625KHz guarding bands on each side. Since we cannot control all 48 QAM points, additional signals are introduced in non-overlapped subcarriers, as shown in Figure 8, which is fine because ZigBee does not sample these subcarriers for demodulation.

To investigate the emulation capacity of source bits further, we increase the total bandwidth of desired subcarriers. When the total bandwidth becomes wider (i.e., the number of elements in Y become larger), the row rank of M increases. Theoretically, when M becomes square and full row-rank, WEBee can control up to 36 subcarriers (216 bits) with a total bandwidth of 11.25MHz, which is much larger than twice the ZigBee bandwidth (2Mhz). It is noted that since we can choose arbitrary elements in Y, WEBee is capable of controlling non-continuous bands of subcarriers, paving the way for parallel CTC.

Scrambler and interleaver: In addition to channel coding, the data scrambling in WiFi OFDM is performed by XORing the incoming source bits with the output of a 7-bit linear feedback shift register. Since the scrambler is a one-to-one mapping from the source bits to the scrambled bits, it is easy to reverse the scrambler by

XORing the scrambled bits with the same output of the shift register once the scrambling seed is known. We can read the scrambling seed from many commodity WiFi radios. For instances, in ath5k supported WiFi cards (e.g., Atheros AR5112 and AR2425), we can fix the scrambling seed by setting the GEN_SCRAMBLER field in the AR5K_PHY_CTL register of the driver. For BCM4330 WiFi chips used by LG Nexus 5, the scrambling seed has been fixed at 8. In other Atheros chipsets (e.g., AR9580 and AR5001G), the scrambling seed is incremented by one between frames, which is easy to track [19]. Similarly, the interleaver is also a one-to-one mapping from the coded bits to the permuted coded bits. This permutation is known in the WiFi standard and can be reversed easily.

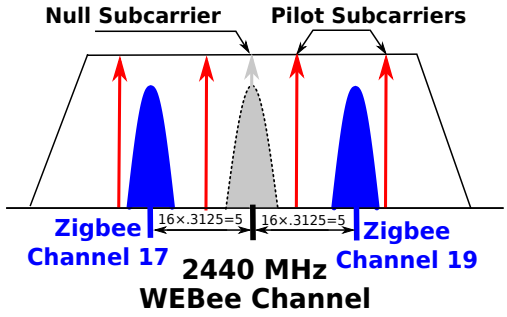


Figure 9: Channels Mapping for Pilot Avoidance

5 POST-QAM EMULATION

This section addresses challenges due to hardware constraints, such as pilot/null subcarrier avoidance, boundary flipping, and cyclic prefixing.

5.1 Pilot/Null Subcarrier Avoidance

In WiFi OFDM, each 20 MHz channel is composed of 64 subcarriers: 48 subcarriers for data, 4 subcarriers as pilot subcarriers for channel state estimation, and 12 null subcarriers. Without hardware modification, the WiFi signals transmitted in the pilot/null subcarrier cannot be controlled by software. Therefore, if the pilot subcarriers overlap with the frequency bands of ZigBee devices, WEBee cannot work properly. In WEBee, channel mapping is necessary to avoid the collision with the pilot/null subcarriers in WiFi OFDM.

Figure 9 shows a channel mapping scheme for pilot subcarrier avoidance. For example, once the central frequency of a WEBee channel is set as 2440MHz, the two regions of WiFi OFDM data subcarriers [-21, -7] and [7, 21]) can be utilized to achieve two parallel WEBee cross-technology communications in standard ZigBee channel 17 and channel 19. We note many commodity WiFi radios (e.g., Atheros AR9485, AR5112, and AR2425) can set their central frequency.

5.2 Post-QAM Challenges

We introduce the high-level idea of QAM evaluation in Section 3.2. In this section, we discuss a few research challenges imposed by the post-QAM process.

Challenge I: Non-continuity after Segmentation: Compared with the 16μ duration of a ZigBee symbol, a WiFi symbol occupies

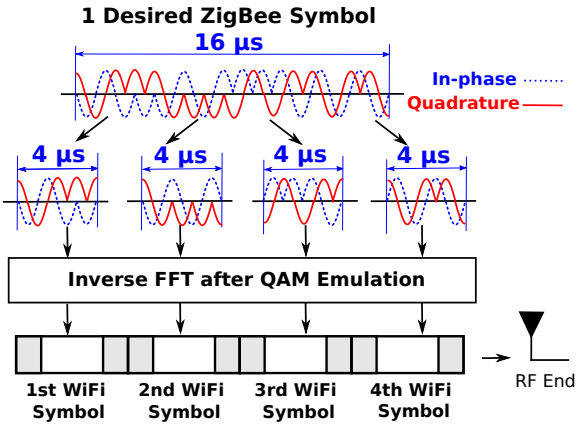


Figure 10: Emulate OQPSK with WiFi QAM.

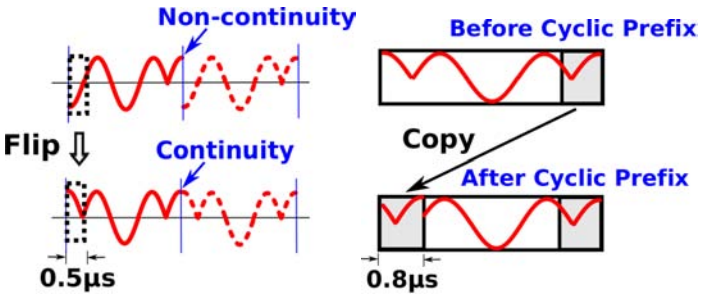
4μs. Therefore, a full ZigBee symbol has to be segmented before emulated by four individual WiFi symbols , as shown in Figure 10. Such segmentation will introduce emulation error without further processing.

Technically, QAM points used by a WiFi symbol are selected by feeding the time-domain signal of one-fourth of a ZigBee symbol into Discrete Fourier transform (DFT). In DFT, the sampling in the frequency domain leads to the replicates of time domain signals; specifically, the signal $\hat{x}(n)$ recovered from the DFT samples via the following synthesis equation

$$\hat{x}(n) = \frac{1}{N}\sum_{k}(\sum_{n} x(n)e^{-j2\pi nk/N})e^{j2\pi kn/N} \quad (4)$$

$\hat{x}(n)$ is a periodic signal that consists of a sum of shifted replicates of the discrete time signal $x(n)$.

On one hand, the DFT treats the time-domain data by assuming they were periodic by concatenating with the shifted replicates of themselves. On the other hand, at the ZigBee receiver, Offset Quadrature Phase-Shift Keying(OQPSK) modulation adds a $\pi/2$ phase offset to the quadrature signals of the ZigBee symbol, which makes concatenated replicates noncontinuous, as shown in Figure 11a. This non-continuity introduces signal boundary errors, when the QAM emulation is used for an individual WiFi symbol, each emulating one-fourth of the *supposedly* continuous ZigBee symbol.



(a) Flipping Effect **(b) Boundary with CP**

Figure 11: Effects of Flipping and Cyclic Prefix

Challenge II: Cyclic Prefixing (CP): Another source of error comes from the WiFi cyclic prefixing, a technique to eliminate inter-symbol interference (ISI). As illustrated in Figure 11b, with cyclic prefixing, a guard interval lasting 0.8μs in each WiFi symbol is copied from the right of WiFi symbol and pasted into (overwrite) the left of the symbol . As a result, the front segment and the end segment of WiFi signals are the same. However, ZigBee I/Q signals do not have such repetition. Because of the cyclic prefix enforced by WiFi modulation, we have a segment of signals with 0.8μs duration which is *out of our control* in signal emulation.

Solution: Selective boundary flipping: Since WEBee does not modify the hardware used, we cannot completely remove these two sources of errors, but we can mediate them. Before feeding one-fourth of the desired ZigBee Q signal to FFT, we build a continuous shifted replicated signal Q' by flipping a half-chip of Q at the boundaries as illustrated in Figure 11a. We note that flipping the desired signal is done in software for the emulation purpose, and thus no hardware changes are involved. Since the duration of a half chip of Q is 0.5μs less than that of a cyclic prefix (i.e., 0.8μs), boundary flipping can mediate errors due to noncontinuity without penalty given the constraint of cyclic prefixing.

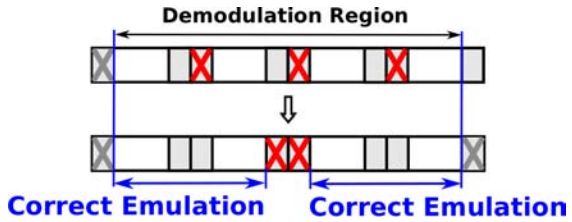


Figure 12: Flipping Boundary Selection

As Figure 12 shows, four WiFi symbols have eight boundaries, among which the left/right-most boundaries are not used by OQPSK demodulation [37]. A naive selection would be flipping only left(right) boundaries of ZigBee signal segments. The upper part of Figure 12 shows the number of regions corrupted by cyclic prefix is 3. But if we choose boundary flipping wisely, as shown in the lower part of Figure 12, the number of regions corrupted by cyclic prefix is reduced to 2, leading to a better demodulation.

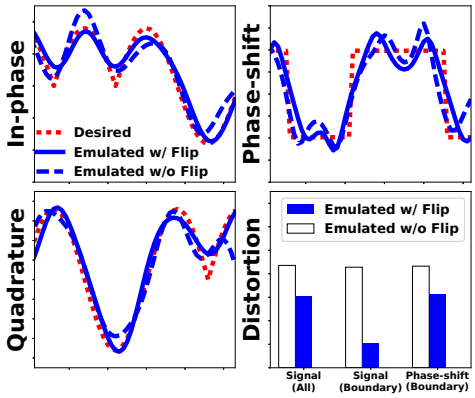


Figure 13: The Effects of Boundary Flip.

Experimentally evaluating our design, Figure 13 shows the emulated signal generated by QAM emulation with selective boundary flip resembles the desired ZigBee signal closely. For example, the phase-shift distortion is significantly lower with selective boundary flip, leading to better chip demodulation performance.

About Robustness of DSSS: It is noted that selective boundary flipping can mediate but not eliminate signal distortion completely. However, thanks to the use of DSSS, the ZigBee symbol can tolerate a certain number of chip errors, leading to a high symbol-level reliability (above 97%), as proven in our testbed later.

6 RELIABLE WEBEE

Due to the intrinsic discrepancy between WiFi and ZigBee, signal distortion cannot be avoided completely during emulation, even with DSSS. Therefore, we need additional high-level mechanisms to achieve highly reliable CTC (e.g., 99% and above).

6.1 Interference and Multipath Effect

In essence, WEBee is a genuine WiFi device. Therefore, WEBee avoids interference from other ISM-band devices (e.g.,WiFi, ZigBee, and BlueTooth) using the standard WiFi CSMA/CA mechanism. The advanced co-existence designs for WiFi [17, 28, 44–46] can be directly applied to WEBee to improve its reliability. Also interestingly, since WEBee emulates the low-bit-rate ZigBee, it can nicely reduce the inter-symbol interference caused by multipath propagation.

6.2 Repeated Transmission for Reliability

For higher reliability, WEBee can simply transmit WiFi frames (i.e., multiple copies of an emulated ZigBee frame) repeatedly. Note that this mechanism is transparent because that a ZigBee receiver cannot tell whether the sender is a WiFi or ZigBee device, and the no special decoding process is required at the ZigBee receiver.

Given the frame reception ratio (FRR), the reception probability of one WEBee frame after m repeated transmission is

$$1 - (1 - FRR)^m. \tag{5}$$

Our later experiment in Section 7.1.3 will show that WEBee's FRR is between 40% and 60%. Accordingly, the reliability is between 95.3% and 99.5% after six retransmissions, as validated by our experiments. Although retransmission is considered inefficient in intra-technology communication, it is acceptable for CTC, given that it is mostly used for low-volume control and coordination. Moreover, in duty-cycled ZigBee networks, it is a common practice to transmit a train of the same frames to wake up the receivers [11, 30].

6.3 Link-layer FEC for Reliability

Beside simple retransmission, WEBee also supports a FEC scheme with two design elements: preamble protection and payload coding. It is not a completely transparent design as retransmission, but it has a better spectrum efficiency with 90 ∼ 99% FRR as shown in evaluation.

6.3.1 Preamble Protection with Repetition. The preamble of an emulated ZigBee frame is a constant string of some symbols '0'

followed by a symbol '7A'. Such frame synchronization cannot be protected by a coding mechanism. WEBee, therefore, provides a repeated frame synchronization mechanism. As shown in Figure 14, preamble repetition improves the chances of successful preamble detection. If the first preamble is identified successfully, WEBee discards the second.

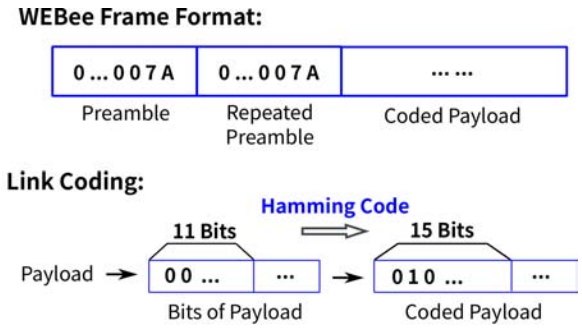


Figure 14: Reliable CTC with Link Coding

6.3.2 Payload Coding. To protect content of an emulated ZigBee frame, we use the computational efficient binary Hamming Code. Since the errors caused by the signal emulation are at the symbol level, while the binary Hamming codes work at the bit level, we employ a simple interleaver scattering the bits in corrupted symbol before applying hamming coding as shown in Figure 14.

The parameter of Hamming Code is determined based on the symbol reception ratio of WEBee, which is higher than 95%. The Hamming Code (15, 11) has the capability of error detection rate of 1/11, which is sufficient to handle the 5% symbol error rate.

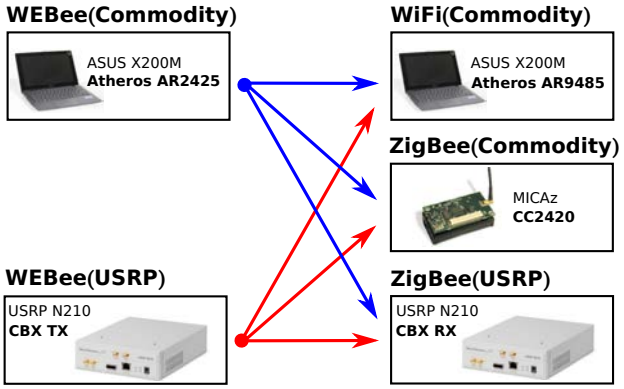


Figure 15: Experiment Setting for WEBee

7 PERFORMANCE EVALUATION

As shown in Figure 15, the WEBee testbed consists of two types of senders: (i) the USRP-N210 platform with 802.11 b/g PHY [3],(ii) a commodity WiFi card Atheros AR2425 as well as three types of receivers: (i) a commodity ZigBee receiver (i.e., MICAz); (ii) a commodity WiFi receiver (i.e., Atheros AR9485); and (iii) a USRP-N210 with 802.15.4 PHY [37].

We note that WEBee CTC is supported directly among commodity devices and USRP-N210 devices are used only for evaluation purposes to measure low-level PHY information, such as the distortion of phase and symbol error rate, which are inaccessible by commodity devices. For the frame reception rate and other high-level metrics, the performance of WEBee in USRP and the WEBee are very similar. With WEBee, a broadcast frame from an Atheros AR2425 WiFi card (or USRP-N210) can be simultaneously received by the MICAz and Atheros AR9485 receivers, indicating this frame is indeed both WiFi-compliant and ZigBee-compliant. It is noted that we also implemented WEBee on LG Nexus 5 smart phones with BCM4330 WiFi chips, indicating our technology is applicable in today's mainstream devices.

During experiments, we fix the scramble seed and configure the mode of transmitter. By adjusting the MTU, each emulated frame consists of a 32-bit preamble, and a default frame payload of 25 bytes, resulting in about a 1ms duration, which is the typical length of payload for ZigBee communications. To ensure statistical validity, we obtain the average result of 10 experiments, each of which sends 1,000 WEBee frames under a wide range of settings including indoor/outdoor, short/long distance, mobile, and duty-cycled scenarios.

7.1 Experimental Results

Our evaluation starts from PHY-layer measurements (i.e., time-domain signal, symbol error rate (SER)) to link-layer statistics (i.e., frame reception ratio (FRR) and throughput). Our study also covers multiple scenarios, including (i) stationary, (ii) long-range, (iii) mobile, (iv) duty-cycled, and (v) parallel CTC.

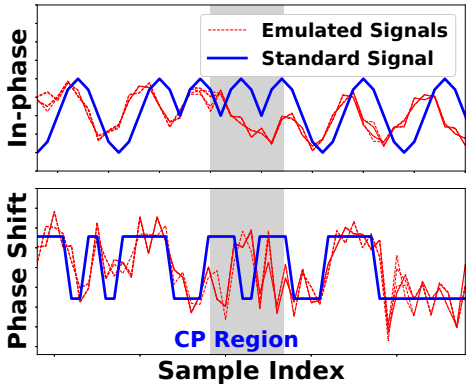


Figure 16: Phase Shifts of Emulated Signals.

7.1.1 Emulated Signals. To ensure reception, the RF waveform of the WiFi payload needs to emulate that of ZigBee signals as closely as possible. It is known that the ZigBee OQPSK demodulation is based on the phase shift, i.e., the angle between two consecutive complex samples. The chip value is decoded as "1" if the phase shift is positive, and otherwise as "0". Figure 16 shows the phase shifts of the received signals by ZigBee (red line) and by ideal standard signal (blue line). Although these two lines are not perfectly aligned, they have the same positive or negative signs, allowing successful chip decoding. After collecting 32 chips, ZigBee

converts these chips to a ZigBee symbol with smallest chip Hamming distances. Using the USRP receiver, we measure the distribution of chip Hamming distances between emulated ZigBee and intended ZigBee symbols. Figure 17 shows that the Hamming distances of emulated symbols are mainly concentrated in the region of [7,12], especially in [8,9].

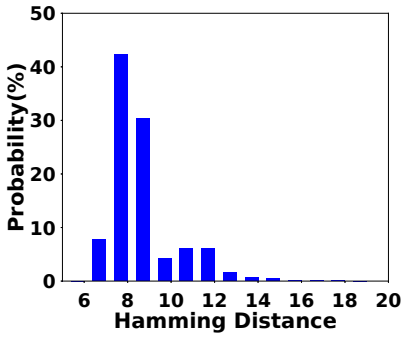


Figure 17: H-Distance of Emulated Signals.

Figure 16 also shows that for all emulated symbols, the central segments of emulated signals always present significant distortions in phase shifts. This phenomenon is a direct result of selective boundary flipping, shown in Figure 12. Because of this emulation constraint, the Hamming distances of emulated signals are always larger than some specific number, such as 6 shown in Figure 17. Thanks to the inherent redundancy in DSSS, such Hamming distances can be tolerated by ZigBee decoding, leading to a low symbol error rate (< 3%).

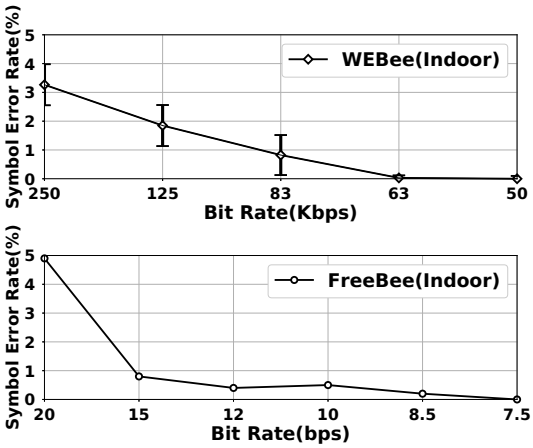


Figure 18: SER of WEBee vs. FreeBee

7.1.2 Symbol Error Rate (SER). Figure 18 shows the main result of this work, comparing the symbol error rate (SER) of WEBee and FreeBee [23]. In WEBee, the symbol is standard ZigBee symbol where each symbol stands for 4 data bits, while the symbol in FreeBee is an interval of consecutive WiFi beacon frames and represents $\lfloor log_2(100) \rfloor = 6$ data bits. The symbol error rates of both WEBee and FreeBee depend on the number of repetition, a parameter to trade off between throughput and reliability. Figure 18 illustrates the SERs under different throughputs: WEBee reaches a 99.9% symbol success rate when the frames are sent at a rate of 63Kbps, while FreeBee achieves a 99.9% symbol success rate with bit rate 7.5bps. Under the same success rate, the throughput of WEBee is more than 8,000 times of that of FreeBee. We note that FreeBee could achieve theoretically 1.4Kbps when it saturates the channel with non-stop beaconing and ignores symbol errors due to interference. We do not compare WEBee with this case as FreeBee is supposed to be a free channel design.

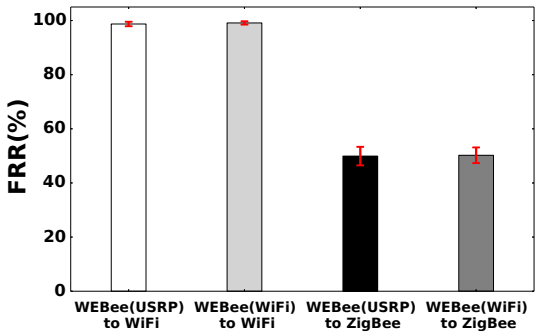


Figure 19: Frame Reception Ratio under Four Settings

7.1.3 Frame Reception Ratio (WiFi vs. ZigBee). WEBee embeds a ZigBee frame into an authentic WiFi frame. To confirm and evaluate such embedding, we conduct experiments using the testbed in Figure 15, where a WEBee sender (either USRP or Atheros AR2425) broadcasts WEBee frames to Atheros AR9485, MICAz CC2420 and USRP N210. Figure 19 shows that the WiFi card Atheros AR9485 receives more than 99% of the frames from both USRP and Atheros AR2425, while the ZigBee CC2420 receives about 50% of the frames.[1] This performance is expected because the WEBee sender indeed transmits a WiFi frame to WiFi receivers, while the WiFi payload cannot emulate ZigBee symbols perfectly. The last bar of Figure 19 shows from the commodity wifi card Atheros AR2425 to the commodity ZigBee CC2420, WEBee achieve around 50% frame reception Ratio (FRR). A similar FRR is achieved using USRP N210 as a sender as shown in the third bar of Figure 19.

Interestingly, this experiment also indicates WEBee can support *cross-technology-broadcast* if we allocate portions of payload to ZigBee and WiFi, separately, because ZigBee naturally ignores non-compliant RF waveforms, while WiFi can decode the whole frame.

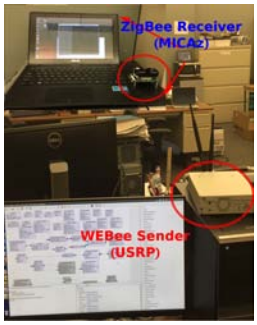


Figure 20: Lab

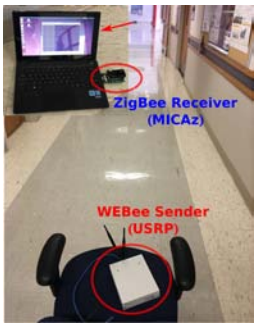


Figure 21: Hallway

[1] As a frame consists of many symbols, a single symbol error would lead to a frame loss.

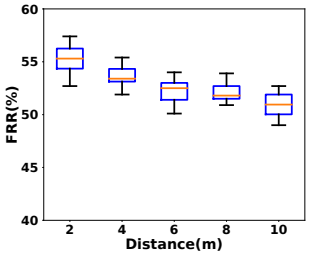


Figure 22: Frame Reception Ratio with Distances (Lab).

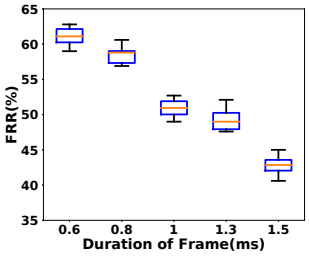


Figure 23: Frame Reception Ratio with Durations (Lab).

7.1.4 WEBee Performance (Indoor vs. Outdoor). We evaluate the indoor performance of WEBee in two sits: (i)a laboratory room (Figure 20) and a hallway (Figure 21). We set the transmission power at 10dBm. Figure 22 shows those frame reception ratios (FRR) between the WEBee sender and the ZigBee receiver under varying distances. With increasing distances from 2 to 10 meters, the FRR drops slowly to 50%, indicating that signal attenuation will not significantly degrade WEBee's reliability.

Figure 23 plots the frame reception ratios with different frame lengths. As expected, the FRRs decrease when the durations of the emulated ZigBee frames increases. Figure 23 also shows that WEBee demonstrates an acceptable performance (>40%) for all the packet lengths that ZigBee normally uses.

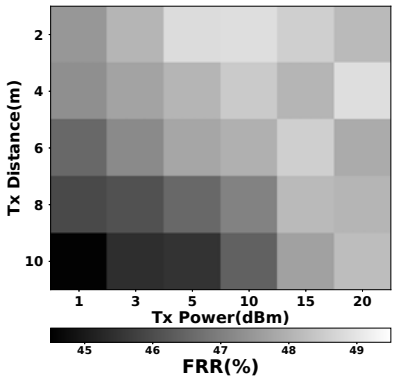


Figure 24: Frame Reception Ratio vs. Tx Powers & Distances

The transmission power also impacts the performance of the signal emulation. Figure 24 shows that the FRRs concentrate in the region between 45% and 55% with different combinations of transmission power and distance settings. When the transmission power of WiFi is set as 20 dBm[2], the FRR of WEBee stays stable at any distance less than 10 meters. This occurs because the signal attenuation has been compensated for by a higher transmission power.

We note that WEBee has a much longer range than that of ZigBee-to-ZigBee communication since WiFi devices normally operate at 20dBm, while ZigBee devices operate at 0dBm. Since energy is less a concern for WiFi devices than for ZigBee devices, this feature is very useful in allowing a WiFi AP to control all IoT devices equipped with low-power ZigBee radios in a large residential area.

[2]the maximum transmitter output power allowed by FCC under ISM band is 30dBm

Figure 25: Outdoor Site

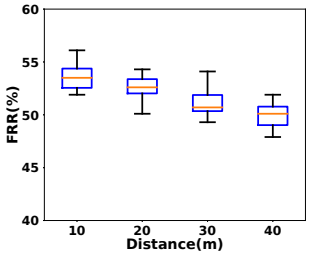


Figure 26: Frame Reception Ratio with Dist. (Hallway)

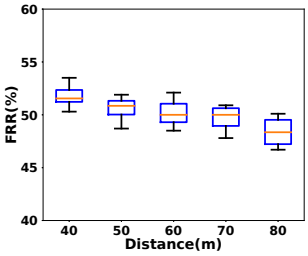


Figure 27: Frame Reception Ratio with Dist. (Outdoor)

To validate, Figures 26 and 27 show the FRR of WEBee with long-distance transmissions at an indoor hallway(Figure 21) and at an outdoor site (Figure 25) where the transmission power are all set as 20dBm. The experimental results show that WEBee achieves the FRRs between 45% and 55% from 10 to 80 meters, which is much further than that of ZigBee-to-ZigBee communication.

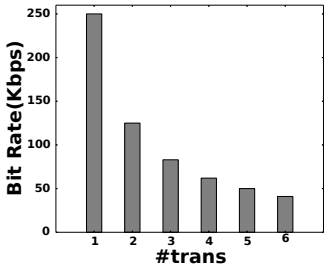


Figure 28: Bit Rates with the number of Trans.

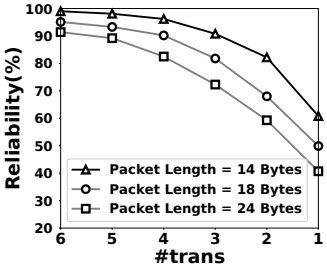


Figure 29: Reliability with the number of Trans.

7.1.5 Repeated Transmission. We show that WEBee can achieve between 45% and 55% FRR with one transmission. To make it reliable enough for practical CTC, we can retransmit the same frame multiple times. We note that retransmission improves the reliability but at the cost of effective throughput, i.e., the bit rate of WEBee CTC drops when #trans become larger. For example, Figure 28 shows that with #trans = 1, WEBee achieves 250Kbps,the bit rate of standard ZigBee, and with #trans = 2, WEBee achieves 125Kbps.

Figure 29 illustrates the reliability of reception with varying #trans and frame lengths. For example, when the emulated frame length is 14 bytes, the reliability of reception is above 99% when the WEBee transmission is repeated 6 times.

The performance of repeated transmission under different distances is also evaluated in Figure 30, where transmission distance is set to 10 meters and frame length is fixed as 18 bytes. From this figure, we can see that repeated transmission has a stable performance when the distance changes.

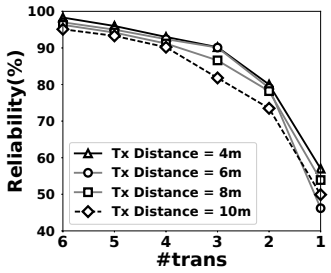


Figure 30: Reliability of Repeated Transmission under Different Tx Dist.

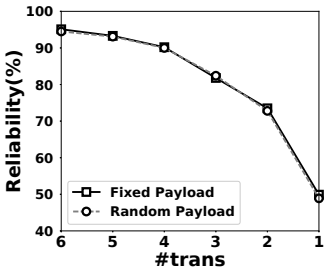


Figure 31: Reliability of Repeated Transmission under Different Payloads.

The impact of different symbols in the emulated ZigBee frames is also evaluated. In the experiment, the transmission distance is set to 10 meters, and the fixed frame length is set to 18 bytes. Figure 31 illustrates the FRRs are very stable, regardless the WiFi frame carry a payload to emulate same ZigBee symbols or a payload to emulate random ZigBee symbols. In summary, repeated transmission is a simple, transparent, and robust mechanism that makes WEBee work well in real wireless environments.

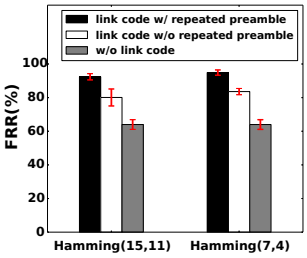


Figure 32: Frame Reception Ratio under 0.5ms Duration

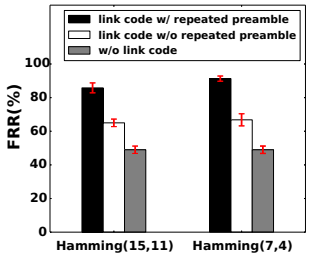


Figure 33: Frame Reception Ratio under 1ms Duration

7.1.6 Link Coding/Decoding. Link coding requires that ZigBee receivers understand the format of link coding, and hence is not as transparent a design as retransmission. On the other hand, WEBee's link coding design can improve reliability more efficiently. Figures 32 and 33 illustrate the performance of WEBee with repeated preamble and link coding. Two types of link coding mechanisms, i.e., Hamming code (15,11) and Hamming code (7,4), are evaluated in this experiment. At the same time, the impact of different frame lengths is also investigated in Figures 32 and 33.

As they show, when the link coding mechanism is utilized, the corruption of a few emulated symbols can be recovered, so the FRR of WEBee is improved shown as Figures 32 and 33. But if the preamble is corrupted, the link coding is not effective. When link coding is combined with repeated preamble, the FRR of WEBee can reach 99% when the frame duration is about 0.5ms, as shown in Figure 32 and above 90% with the 1ms frame duration.

The difference between Hamming (7,4) and Hamming (15,11) is very small if the frame length is short, and such a difference would slightly increase when the frame length becomes larger. It is the effect of low symbol error rate in WEBee, the Hamming (15,11) has enough capability to recover such errors.

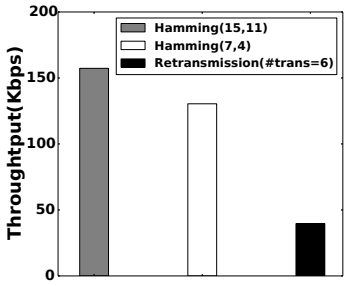


Figure 34: Throughput Coding vs. Retransmission

Furthermore, Figure 34 compares the throughput of the link coding and retransmission. As shown, the link coding obtains a high reliability with good throughput while retransmission only gain a high reliability with the cost of decreasing the effective throughput. This is a tradeoff between the receiver transparency and throughput.

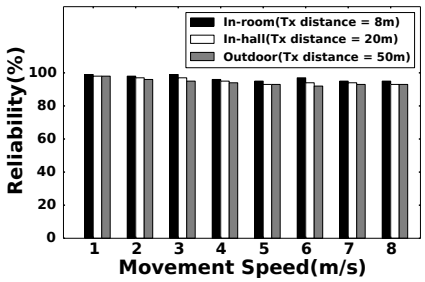


Figure 35: Reliability under Mobility.

7.1.7 Performance under Mobility. We also evaluate WEBee with a mobile ZigBee receiver at three sites (i.e., room, hallway, and outdoor). In this experiment, each WEBee frame is retransmitted 8 times and the interval between consecutive frames is set as 8ms. The speeds are set between 1m/s and 8m/s, respectively. Figure 35 shows that WEBee works well under different wireless environments and different levels of mobilities.

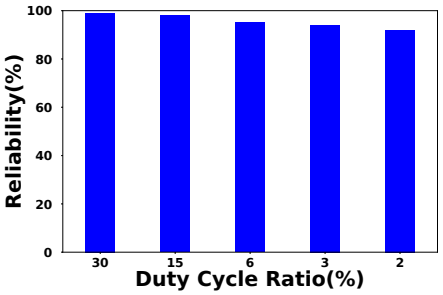


Figure 36: Reliability under Low Duty-Cycle.

7.1.8 Performance under Low Duty-Cycles. WEBee is also evaluated using low-duty-cycle ZigBee receivers, which sleep most of the time and wake up periodically to check for radioactivity. In this case, a sender needs to transmit a train of repeated frames to wake up the receiver [11, 30]. In our experiments, each WEBee frame is retransmitted 100 times. The duty-cycle ratios are set as 30%, 15%, 6%, 3%, and 2%, respectively. The results in Figure 39 show that the WEBee can work well under a wide range of duty-cycle settings.

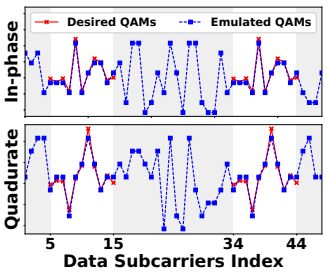


Figure 37: QAM Points in Data Subcarriers for Parallel Communication.

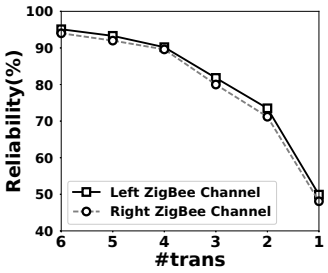


Figure 38: Frame Reception Ratio on Different Parallel ZigBee Channels.

7.1.9 Parallel Communication. Here we show that WEBee can support two parallel communications from WiFi to ZigBee. Using the channel mapping, as shown in Figure 9, WEBee sends two ZigBee frames with two different subcarrier regions (11 data subcarriers each) in WiFi as shown in Figure 37. Figure 38 compares the performance of WEBee between the left and right ZigBee channels when repeated transmission is utilized. In this experiment, the transmission distance is 10 meters, and the frame length is 18 bytes.

Figure 38 shows that the FRRs of WEBee are above 93% under different parallel ZigBee channels. At the same time, the FRR at one channel is slightly worse than the FRR at another channel overall. The reason for this phenomenon is the diversity of the channel quality under different channels. With two parallel channels, the aggregate throughput of WEBee can be more than 16,000x faster than that of packet-level CTCs.

Figure 39: WEBee Smart Light Control

7.1.10 Application: Smart Light Control. The last experiment shows that WEBee can be used in real-life settings. We embed WEBee into an LG Nexus 5 smart phone equipped with a Broadcom BCM4330 WiFi chip. Without making any hardware modification on either side, we tunnel Zigbee control packets through the WEBee link to control light intensity, color and on/off status of ZigBee bulbs. All control operations were successful during the experiments. Our solution eliminates the need for expensive ZigBee gateways (e.g., WINK Hub), hence reduces the cost of light control in smart environments. The related videos can be found at [24, 25] and the technical support for WEBee is available at [10].

8 RELATED WORKS

In its early days, research on wireless coexistence focused mostly on cross-technology inference avoidance, cancellation, and detection [1, 2, 6–8, 16, 21, 29, 31, 34–36, 38–40, 45–48, 50–52]. Recently, cross-technology communications (CTC) have emerged as a mechanism for explicit coordination and collaboration. FreeBee[23] establishes CTC by modulating the interval of WiFi Beacons. Esense [5] establishes communication channels from WiFi to ZigBee by modulating the lengths of WiFi frames. HoWiES [49] extends the Esense mechanism to convey data with combinations of WiFi frames. GSense [48] uses gaps embedded between the customized preamble to deliver cross-technology communications. In comparison with these packet-level CTC [5, 9, 23, 48, 49], WEBee is the first work to implement the CTC based on physical signal emulation, leading to 16,000x throughput improvement. Also, WEBee is the first work to achieve parallel CTC, which is infeasible using packet-level modulations (e.g., RSSI and timing).

The WEBee's concept of signal emulation is inspired by several recent works [4, 19, 22, 26, 32] that implement the signal manipulation. The research in [22] produces standard WiFi signals by controlling the backscatter. The work in [19] provides an amplitude modulated pulse signal that could be detected by RFID using WiFi devices. The work in [4] generates WiFi CTS frame from the LTE devices, which is no longer a LTE-complaint MAC frame.

Uniquely, WEBee has the capability to emulate ZigBee signals with WiFi payloads without any hardware and firmware modification. Its dual-standard compliance (i.e., ZigBee and WiFi) enables many interesting designs, such as WiFi-ZigBee dual payloads for cross-technology broadcast, cross-technology-encapsulation (i.e., encapsulating ZigBee frames for multi-hop networked control over IP). These features are not supported in existing works [4, 19, 22].

9 CONCLUSION

This work presents WEBee, a physical-level cross-technology communication design based on signal emulation. Using multiple subcarrier regions, WEBee is the first to offer parallel cross-technology communication. Our extensive experiments show that WEBee achieves bit rates more than 16,000x those of the current state of the art with 99% symbol reliability.

Our experiments also demonstrate that WEBee can work well under long-distance, mobile, and low duty-cycle scenarios. As future work, our recent development indicates that enhanced versions of WEBee can support bi-directional communication and the MIMO OFDM modulation under 802.11n, offering more opportunities for high-level cross-technology coordination in the ISM band. Last, we note that the vision of Software Defined Radio (SDR) allows a single transceiver to adapt as needed, but its pure software-based design requires significant cost, energy, and computational complexity. WEBee, in contrast, opens a new pathway to achieve SDR through emulation, striking a nice balance among reliability, flexibility, deployability, and complexity.

ACKNOWLEDGEMENT

This work was supported in part by the NSF CNS-1444021, NSF CNS-1718456 and NSF China 61672196. We sincerely thank our shepherd Shyam Gollakota and anonymous reviewers for their valuable comments and feedback.

REFERENCES

[1] Fadel Adib, Swarun Kumar, Omid Aryan, Shyamnath Gollakota, and Dina Katabi. 2013. Interference alignment by motion. In *MobiCom '13*. ACM, 279–290.

[2] Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, Rohan Murty, and Matt Welsh. 2009. White space networking with wi-fi like connectivity. *ACM SIGCOMM Computer Communication Review* 39, 4 (2009), 27–38.

[3] B. Bloessl, M. Segata, C. Sommer, and F. Dressler. 2013. An IEEE 802.11a/g/p OFDM Receiver for GNU Radio. In *In ACM SIGCOMM 2013, 2nd ACM SIGCOMM Workshop of Software Radio Implementation Forum (SRIF 2013)*.

[4] Eugene Chai, Karthik Sundaresan, Mohammad A Khojastepour, and Sampath Rangarajan. 2016. LTE in unlicensed spectrum: are we there yet?. In *MobiCom '16*. ACM, 135–148.

[5] Kameswari Chebrolu and Ashutosh Dhekne. 2009. Esense: Communication through energy sensing. In *MobiCom '09*. ACM, 85–96.

[6] Bo Chen, Yue Qiao, Ouyang Zhang, and Kannan Srinivasan. 2015. Airexpress: Enabling seamless in-band wireless multi-hop transmission. In *MobiCom '15*. ACM, 566–577.

[7] Bo Chen, Vivek Yenamandra, and Kannan Srinivasan. 2015. Interference alignment using shadow channel. In *INFOCOM 2015*. IEEE, 2128–2136.

[8] Lin Chen, Ruolin Fan, Kaigui Bian, Mario Gerla, Tao Wang, and Xiaoming Li. 2015. On heterogeneous neighbor discovery in wireless sensor networks. In *INFOCOM '15*. IEEE, 693–701.

[9] Zicheng Chi, Yan Li, Hongyu Sun, Yao Yao, Zheng Lu, and Ting Zhu. 2016. B2W2: N-Way Concurrent Communication for IoT Devices. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 245–258.

[10] "The Minnesota Networks Data and System Group". 2017. Cross-Technology-Communication. (2017). Available at http://tianhe.cs.umn.edu/CTC.

[11] Adam Dunkels. 2011. The contikimac radio duty cycling protocol. (2011).

[12] FCC. 2014. Increased Availability of Spectrum for Unlicensed Uses in the 5 GHz Band. (2014). Available at https://www.fcc.gov/document/fcc-increases-5ghz-spectrum-wi-fi-other-unlicensed-uses.

[13] FCC. 2015. Rules for unlicensed operations in the TV and the 600 MHz band. (2015). Available at https://www.fcc.gov/document/fcc-adopts-rules-unlicensed-services-tv-and-600-mhz-bands.

[14] FCC. 2016. *Fact Sheet: Spectrum Frontiers Order To Identify, Open Up Vast Amounts Of New High-Band Spectrum For Next Generation (5g) Wireless Broadband*. Available at https://apps.fcc.gov/edocs_public/attachmatch/DOC-340310A1.pdf.

[15] Inc Gartner. 2016. *Gartner Report*. Available at http://cloudtimes.org/2013/12/20/gartner-theinternet-of-things-will-grow-30-times-to-26-billion-by-2020/.

[16] Tian Hao, Ruogu Zhou, Guoliang Xing, Matt W Mutka, and Jiming Chen. 2014. Wizsync: Exploiting wi-fi infrastructure for clock synchronization in wireless sensor networks. *IEEE Transactions on mobile computing* 13, 6 (2014), 1379–1392.

[17] A. C. Hsu, D. S. L. Wei, and C. C. J. Kuo. 2015.. Coexistence WiFi MAC design for mitigating interference caused by collocated bluetooth.. In *IEEE Trans. Computers, 64(2):342-352,*.

[18] Texas Instruments. 2006.. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver.

[19] Vikram Iyer, Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua Smith. 2016. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference*. ACM, 356–369.

[20] Wenchao Jiang, Zhimeng Yin, Song Min Kim, and Tian He. 2017. Transparent Cross-technology Communication over Data Traffic. In *INFOCOM 2017*.

[21] Tao Jin, Guevara Noubir, and Bo Sheng. 2011. Wizi-cloud: Application-transparent dual zigbee-wifi radios for low power internet access. In *INFOCOM, 2011 Proceedings IEEE*. IEEE, 1593–1601.

[22] Bryce Kellogg, Vamsi Talla, Shyamnath Gollakota, and Joshua R Smith. 2016. Passive wi-fi: Bringing low power to wi-fi transmissions. In *NSDI '16*. USENIX Association, 151–164.

[23] Song Min Kim and Tian He. 2015. FreeBee: Cross-technology Communication via Free Side-channel. In *MobiCom '15 (MobiCom '15)*. ACM, New York, NY, USA, 317–330. https://doi.org/10.1145/2789168.2790098

[24] Zhijun Li, Ling Liu, and Tian He. 2017. WEBEE Demo and Smartligh Control. (2017). Available at https://youtu.be/eHVDn20FOqM.

[25] Zhijun Li, Ling Liu, and Tian He. 2017. WEBee Smartligh Control. (2017). Available at https://youtu.be/Nf9ng1E2Dj0.

[26] Zhenjiang Li, Yaxiong Xie, Mo Li, and Kyle Jamieson. 2015. Recitation: Rehearsing wireless packet reception in software. In *MobiCom '15*. ACM, 291–303.

[27] Zhijun Li, Zhimeng Yin, Ruofeng Liu, Ling Liu, and Tian He. 2017. Demo: WEBee: Physical-Layer Cross-Technology. In *MobiCom 2017*.

[28] Chieh Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. 2010. Surviving WiFi Interference in Low Power ZigBee Networks. In *SenSys '10*.

[29] Rajesh Mahindra, Hari Viswanathan, Karthik Sundaresan, Mustafa Y Arslan, and Sampath Rangarajan. 2014. A practical traffic management system for integrated LTE-WiFi networks. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 189–200.

[30] David Moss and Philip Levis. 2008. BoX-MACs: Exploiting physical and link layer boundaries in low-power networking. (2008).

[31] Georgios Nikolaidis, Mark Handley, Kyle Jamieson, and Brad Karp. 2015. COPA: cooperative power allocation for interfering wireless networks. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*. ACM, 18.

[32] Jiajue Ou, Yuanqing Zheng, and Mo Li. 2014. MISC: Merging incorrect symbols using constellation diversity for 802.11 retransmission. In *INFOCOM 2014*. IEEE, 2472–2480.

[33] Qualcomm Technologies, Inc. 2014. *LTE Advanced in unlicensed spectrum*. Qualcomm Technologies, Inc. Available at http://www.qualcomm.com/solutions/wireless-networks/technologies/lte-unlicensed.

[34] Božidar Radunović, Ranveer Chandra, and Dinan Gunawardena. 2012. Weeble: Enabling low-power nodes to coexist with high-power nodes in white space networks. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 205–216.

[35] Saravana Rathinakumar, Bozidar Radunovic, and Mahesh K Marina. 2016. CPRecycle: Recycling Cyclic Prefix for Versatile Interference Mitigation in OFDM based Wireless Systems. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*. ACM, 67–81.

[36] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Ranveer Chandra, and Jie Liu. 2016. SNOW: Sensor network over white spaces. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*.

[37] T Schmid. 2005. *Gnu radio 802.15. 4 en-and decoding*. Technical report, UCLA NESL (2005).

[38] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. 2011. No time to countdown: Migrating backoff to the frequency domain. In *MobiCom '11*. ACM, 241–252.

[39] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. 2013. Successive interference cancellation: Carving out MAC layer opportunities. *IEEE Transactions on Mobile Computing* 12, 2 (2013), 346–357.

[40] Karthikeyan Sundaresan, Srikanth V Krishnamurthy, Xinyu Zhang, Amir Khojastepour, Sampath Rangarajan, et al. 2015. TRINITY: A Practical Transmitter Cooperation Framework to Handle Heterogeneous User Profiles in Wireless Networks. In *MobiHoc '15*. ACM, 297–306.

[41] Wi-Fi Alliance. 2016. Wi-Fi HaLow. (2016). Available at http://www.wi-fi.org/discover-wi-fi/wi-fi-halow.

[42] Eric W. Weisstein. 2000. Parseval's Theorem. (2000). Available at http://mathworld.wolfram.com/ParsevalsTheorem.html.

[43] Zhimeng Yin, Wenchao Jiang, Song Min Kim, and Tian He. 2017. C-Morse: Cross-technology Communication with Transparent Morse Coding. In *INFOCOM 2017*.

[44] Sangki Yun, Daehyeok Kim, and Lili Qiu. 2013. Fine-grained spectrum adaptation in wifi networks. In *MobiCom '13*. ACM, 327–338.

[45] Sangki Yun and Lili Qiu. 2015. Supporting WiFi and LTE co-existence. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 810–818.

[46] Xinyu Zhang and Kang G Shin. 2011. Enabling coexistence of heterogeneous wireless systems: Case for ZigBee and WiFi. In *MobiHoc '11*. ACM, 6.

[47] Xinyu Zhang and Kang G Shin. 2013. Cooperative carrier signaling: Harmonizing coexisting WPAN and WLAN devices. *IEEE/ACM Transactions on Networking* 21, 2 (2013), 426–439.

[48] Xinyu Zhang and Kang G Shin. 2013. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *INFOCOM, 2013 Proceedings IEEE*. IEEE, 3094–3101.

[49] Yifan Zhang and Qun Li. 2013. HoWiES: A holistic approach to ZigBee assisted WiFi energy savings in mobile devices. In *INFOCOM, 2013 Proceedings IEEE*. IEEE, 1366–1374.

[50] Ruogu Zhou, Yongping Xiong, Guoliang Xing, Limin Sun, and Jian Ma. 2010. ZiFi: wireless LAN discovery via ZigBee interference signatures. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 49–60.

[51] Wenjie Zhou, Tarun Bansal, Prasun Sinha, and Kannan Srinivasan. 2014. Bbn: throughput scaling in dense enterprise wlans with bind beamforming and nulling. In *MobiCom '14*. ACM, 165–176.

[52] Wenjie Zhou, Tanmoy Das, Lu Chen, Kannan Srinivasan, and Prasun Sinha. 2016. BASIC: backbone-assisted successive interference cancellation. In *MobiCom '16*. ACM, 149–161.