WiBeacon: Expanding BLE Location-based Services via WiFi

Ruofeng Liu
University of Minnesota, Twin Cities
Minneapolis, MN, USA
liux4189@umn.edu

Zhimeng Yin
City University of Hong Kong
Hong Kong SAR, China
zhimeyin@cityu.edu.hk

Wenchao Jiang
Singapore University of Technology and Design
Singapore
wenchao_jiang@sutd.edu.sg

Tian He
University of Minnesota, Twin Cities
Minneapolis, MN, USA
tianhe@umn.edu

ABSTRACT

Despite the popularity of Bluetooth low energy (BLE) location-based services (LBS) in Internet of things applications, large-scale BLE LBS are extremely challenging due to the expenses of deploying and maintaining BLE beacons. To alleviate this issue, this work presents WiBeacon, which repurposes ubiquitously deployed WiFi access points (AP) into virtual BLE beacons via only moderate software upgrades. Specifically, a WiBeacon-enabled AP can broadcast elaborately designed WiFi packets that could be recognized as iBeacon-compatible location identifiers by unmodified mobile BLE devices. This offers fast deployment of BLE LBS with *zero* additional hardware costs and *low* maintenance burdens. WiBeacon is carefully integrated with native WiFi services, retaining transparency to WiFi clients. We implement WiBeacon on commodity WiFi APs (with various chipsets such as Qualcomm, Broadcom, and MediaTek) and extensively evaluate it across various scenarios, including a real commercial application for courier check-ins. During the two-week pilot study, WiBeacon provides reliable services, i.e., as robust as conventional BLE beacons, for 697 users with 150 types of smartphones.

CCS CONCEPTS

• **Networks → Wireless local area networks**.

KEYWORDS

Cross-technology Communication, Bluetooth Low Energy, Wi-Fi, Localization

ACM Reference Format:
Ruofeng Liu, Zhimeng Yin, Wenchao Jiang, and Tian He. 2021. WiBeacon: Expanding BLE Location-based Services via WiFi. In *The 27th Annual International Conference On Mobile Computing And Networking (ACM MobiCom '21), October 25–29, 2021, New Orleans, LA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3447993.3448615

ACM MobiCom '21, October 25–29, 2021, New Orleans, LA, USA

ACM ISBN 978-1-4503-8342-4/21/10…$15.00
https://doi.org/10.1145/3447993.3448615

1 INTRODUCTION

Bluetooth Low Energy (BLE) location-based services (LBS) have been widely deployed in various Internet-of-things (IoT) scenarios (e.g., retail stores [42], sports stadiums [55], and airports [35]). More recently, BLE LBS with million+ users have been launched. To enable instant delivery (e.g., real-time courier tracking and order scheduling), a city-wide BLE system was deployed at more than 10,000 restaurants in Shanghai, while the future nationwide deployment is expected to involve 2.5 million merchants [27]. Besides, BLE LBS are also being extensively used in automatic check-ins for contact tracing during pandemics (e.g., COVID-19) [5].

Large-scale deployments of BLE location services, however, are extremely challenging. Conventionally, BLE infrastructures (i.e., BLE beacons [6]) need to be massively installed and configured to broadcast BLE location identifiers. A large number of dedicated beacon devices inevitably incur tremendous deployment costs and long-term maintenance burdens. According to a recent report from Forrester Research, each beacon costs around $300 per year [28].

In this paper, we present WiBeacon, a low-cost solution for these emerging large-scale BLE LBS that require moderate positioning accuracy (e.g., city-wide check-ins and contact tracing [27]). Specifically, inspired by recent advances in *cross-technology communication* (CTC) [31, 44, 48] that enable direct communication among heterogeneous wireless protocols, we propose to repurpose *already deployed* WiFi infrastructures to be virtual BLE beacons. By software upgrades on existing WiFi access points (AP), they can broadcast BLE location identifiers to mobile devices, achieving several advantages. Firstly, benefiting from ubiquitous installations of WiFi APs over the last 10 years, WiBeacon enables rapid deployments of BLE location services without incurring additional hardware costs. Secondly, built-in Internet connectivity of APs allows system administrators to monitor, manage, and upgrade a large number of virtual beacons remotely, thus significantly reducing long-term maintenance burdens of beacons that are geographically scattered.

Admittedly, the idea of CTC has been extensively discussed in previous literature [31, 44, 48]. However, previous designs only studied the compatibility among heterogeneous wireless devices at the *physical layer* (i.e., enable the communication from WiFi to BLE without hardware modification of either side). In contrast, WiBeacon targets building a full-fledged location service. Therefore, we have to move from previous physical-layer compatibility to full compatibility at the *service level* of both sides. First, our BLE

location service must be strictly compatible with both hardware and software of mobile devices requiring *zero* modification. Second, our software upgrades of the AP should also be compatible with native WiFi services and transparent to WiFi clients. More specifically, we need to address two critical issues:

How can we avoid modifying mobile devices? Existing CTC techniques require receivers (i.e., mobile devices) to make significant modifications, which renders the service incompatible with billions of unmodified devices. This is because WiFi radios suffer from hardware restrictions and thus are inhibited from perfectly generating standard wireless signals following the existing BLE LBS protocols (e.g., iBeacon [2]). To overcome communication errors, previous designs add extra error correction contents into packets, thus violating the standard packet structures of existing protocols. Therefore, mobile devices must be modified in advance to interpret the message. However, it is impractical to modify all potential devices on large-scale LBS.

How can we integrate BLE LBS with the native WiFi service? A WiFi AP normally serves clients on one single WiFi channel, whereas BLE LBS (e.g., iBeacon) requires periodic frequency hopping to increase the detection probability. Without a careful integration of two services, WiFi networks may suffer from severe performance degradation (e.g., transmission loss or even unexpected disconnection). Since previous designs mostly target at the physical-layer compatibility, it is still an open issue how to accomplish the service-level integration without sacrificing WiFi networks.

To the best of our knowledge, WiBeacon is the *first* cross technology design that achieves *service-level* compatibility with both mobile devices and the WiFi AP. A WiBeacon-enabled AP provides BLE location services strictly following iBeacon protocol, thus requiring zero modification of mobile devices. Furthermore, WiBeacon service is seamlessly integrated with the native WiFi service of the AP, thus being fully transparent to WiFi clients. More specifically, our technical highlights are as follows:

iBeacon-compatible Service for Unmodified Mobile Devices: WiBeacon manages to broadcast elaborately designed WiFi packets that can be identified by unmodified mobile devices as standard iBeacon broadcasts with zero errors. To achieve this, we address the most fundamental barrier for cross-technology designs to fulfill compatibility with the existing protocol: *signal imperfection*. Our key technical insight is that although the transmitted signal is inevitably imperfect due to hardware restrictions of a WiFi transmitter, we can elaborately exploit low-pass filter (LPF), a standard component of BLE radios, to mitigate imperfections at the receiver side. Based on this critical insight, we propose a novel approach to generate WiFi signals with unique imperfection patterns, which can take advantage of LPF to effectively eliminate communication errors without making any modification to mobile devices.

WiFi-compatible Integration on Commodity APs: We meticulously integrate WiBeacon into commodity WiFi APs while retaining compatibility with the existing WiFi service. Specifically, we carefully reuse several features of 802.11 standards to emulate the frequency hopping of iBeacon. By doing so, we ensure that our integration is compliant with 802.11 standards, transparent to WiFi clients, and applicable to massive numbers of APs with different WiFi chipsets. Besides, we present a dynamic scheduling algorithm to minimize WiBeacon's impact on WiFi network performances.

We implement WiBeacon on OpenWrt [12] and evaluate it across WiFi APs using Qualcomm, Broadcom, MediaTek chipsets. Extensive experiments show that it achieves compatibility with both mobile devices and native WiFi services.

To further validate the practicality of WiBeacon in the complex real-world scenario, we deploy it in a *real* commercial LBS application, i.e., meal courier tracking in intelligent food delivery. Specifically, we cooperate with an instant delivery company and upgrade the existing APs of restaurants to serve as BLE LBS infrastructures. We envision that WiBeacon will dramatically cut down hardware costs and administrative burdens. During our two-week pilot study, WiBeacon provides location services for 697 meal couriers and helps to track 1780 food orders. The results of this real-world pilot study demonstrate that WiBeacon offers as reliable services as conventional BLE beacons.

In summary, our intellectual contributions are as follows:

- We propose WiBeacon - a low-cost solution for large-scale BLE location services by reusing existing WiFi infrastructure at the cost of *zero* additional hardware and *little* remote configuration.
- WiBeacon addresses several technical challenges in cross-technology designs to achieve strict service-level compatibility with unmodified mobile devices and seamless integration with WiFi services.
- We integrate WiBeacon into COTS WiFi APs. The software and detailed instructions for making WiFi iBeacon-compliant are provided in [11] for the community to reproduce our experiments.
- We extensively validate it in the real commercial LBS application with 150 types of smartphones.

2 MOTIVATION

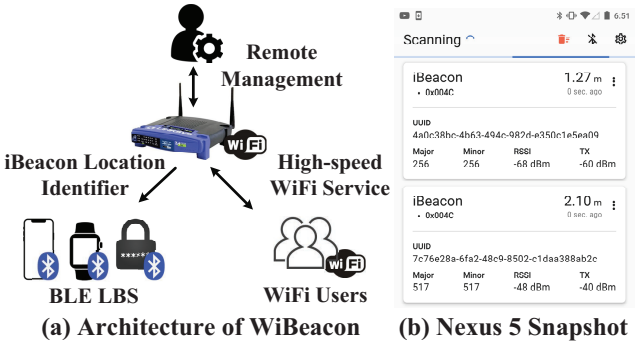
This section presents the motivation of reusing ubiquitous WiFi APs for BLE location-based services.

Why are Large-scale Deployments of BLE Beacon Challenging? Conventionally, BLE location services require dedicated beacons that broadcast their identifiers to nearby mobile devices to indicate the proximity. Although effective, these stand-alone, battery-powered beacons suffer from high deployment and administrative costs when they are deployed on a large scale. In the deployment stage, users need to exert a lot of effort purchasing beacon devices, manually installing them in situ, and labeling their locations in the maps, which may take a few months [17]. After the installation, the management of beacons is even more labor-intensive. Even with one-year battery life, a thousand beacons would lead to a large number of replacements every week due to beacon loss in building renovations, battery drain, hardware errors, etc. [16]. What makes it worse is that since beacons are typically stand-alone without Internet connections, the failures of beacons are hard to detect, and thus extensive site visits are required for maintenance. A Forrester Research report estimates that each beacon costs around $300 per year [28], incurring non-trivial expenses on massive deployments.

These fundamental limitations of dedicated beacons motivate us to explore a cost-effective alternative for BLE LBS.

Why cannot WiFi LBS Replace iBeacon? One may wonder if we can directly use WiFi protocol for LBS. Despite being extensively studied in the literature, WiFi LBS also has several practical limitations. First, contrary to the universal accessibility of iBeacon data on all mobile platforms, the list of scanned WiFi APs are not accessible to developers on some mobile OS (e.g., IOS [29]) due to security concerns. This inhibits WiFi LBS from being adopted in commercial applications that must be compatible with any potential mobile devices. Second, WiFi scans incur high power consumption to mobile devices, which drains their battery fast. In specific, WiFi reception consumes 600mW [52], significantly higher than BLE (30mW) [36]. The measurement study [47] shows that frequent WiFi scan reduces the battery life by up to 90%. As a result, mobile devices typically restrict WiFi scan interval to 60 s with screen on and 300 s with screen off [53], which introduces unacceptable latency to location services that commonly desire real-time performance. Finally, there are a lot of low-cost smart devices and wearables that are only equipped with BLE chipsets and do not work with WiFi. For instance, BLE-enabled smart lock in shared bikes can only detect the entry to geo-fences via receiving iBeacon broadcasts [38].

Benefits of BLE LBS via Ubiquitous WiFi APs: We propose to reuse deployed WiFi APs to provide BLE LBS, which combines advantages of both BLE and WiFi, i.e., the well-developed commercial iBeacon ecosystem and ubiquitous WiFi infrastructures. In specific, 432 million public WiFi access points deployed in the last decade [56] enable the large-scale deployment of BLE infrastructures with little effort and in a short time. Furthermore, since the access points are connected to the Internet and power supply, they allow remote management and get rid of the need for battery changes, which significantly reduce maintenance costs. In the meantime, all the valuable features of iBeacon ecosystem are inherited, including universal compatibility across every mobile platform, thousands of developed iBeacon applications, and ultra-low power consumption of mobile devices in real-time location-based applications.



(a) Architecture of WiBeacon (b) Nexus 5 Snapshot

Figure 1: WiBeacon Turns Ubiquitous WiFi APs into Virtual BLE beacons for Location-based Services.

3 OVERVIEW

In a nutshell, WiBeacon turns WiFi APs into virtual BLE beacons, providing BLE location services with ubiquitous WiFi infrastructure. As Fig.1(a) depicts, the system administrator upgrades and configures WiBeacon software on a deployed WiFi AP remotely via an Internet connection. The WiFi AP periodically broadcasts legitimate WiFi frames with elaborately selected payloads (Section §5). These WiFi frames that can be recognized by unmodified mobile devices in the same way of discovering standard iBeacon location broadcasts. Fig.1(b) shows a snapshot of iBeacon scanner in Nexus 5 with unique location identifiers (e.g., UUID) and the estimated proximity of WiBeacon. WiBeacon is carefully integrated with WiFi services (Section §6) so that the AP maintains high-speed services (e.g., 802.11g/n/ac/ax) for WiFi clients.

4 BACKGROUND

This section provides the background of iBeacon protocol and analyzes the limitations of previous cross-technology communication (CTC) designs.

4.1 iBeacon Preliminary

An iBeacon-compatible beacon emits BLE broadcasts with the format depicted in Fig.2, which is modulated using BLE 4.x 1 Mbps Gaussian Frequency Shift Key (GFSK) and transmitted at BLE advertising channels. Specifically, each data bit is modulated to one BLE chip, which is either a positive or a negative phase shift of 1 us duration. Note that BLE 4.x does not provide any redundant chips for error corrections. Consequently, an iBeacon-compatible service is fundamentally required to broadcast with **zero chip error**.

A mobile device captures the broadcast signal at BLE advertising channels and passes it through a 1 MHz low pass filter (LPF). The filtered signal is demodulated via quadrature demodulation. In specific, the receiver first samples the waveform at 1 MHz (T_0, ..., T_n). The changes of phase, i.e., phase shifts between consecutive complex I/Q samples are then calculated. Finally, the receiver interprets the sign of phase shifts into bits: positive phase shifts are decoded as bit 1 while negative ones are decoded as bit 0. Decoded frames without chip error are used by mobile devices for proximity estimation, whereas a frame with any chip error is discarded.

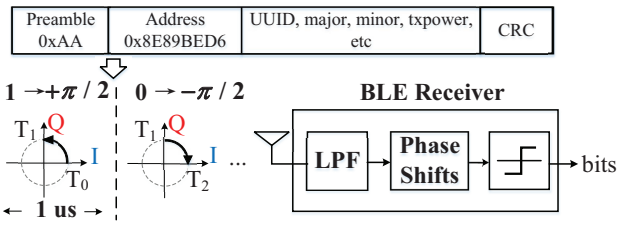


Figure 2: Frame Format, Modulation and Demodulation of an iBeacon Broadcast Frame.

4.2 Limitation of Existing CTC

Cross-technology communications (CTC)[22, 31, 48, 49] enable WiFi radios to emulate heterogeneous wireless signal. However, the emulated signal inherently suffers from signal imperfections due to hardware restrictions of a WiFi radio [31]. Due to the imperfections, existing designs cannot produce iBeacon broadcasts without chip error and thus fails to be strictly compatible with unmodified mobile devices.

Specifically, existing designs exclusively use frequency division multiplexing (OFDM) modulator in WiFi radios to produce heterogeneous signal (thus named "OFDM method"). As the left part of Fig.3 depicts, each OFDM symbol has a cyclic prefix (CP), so the first 0.8 μs signal of an 4 μs OFDM symbol has to be exactly the same as the signal at the last 0.8 μs. In contrast, BLE signals (depicted in the right part of Fig.3) do not have such repetition. Due to this hardware restriction, one out of every 4 BLE chips is significantly distorted, which leads to severe chip errors. Although corrupted signals might still be detected and decoded by a BLE radio (i.e., physical-layer compatible), the result will be ignored by unmodified mobile devices and thus cannot be used for location services.

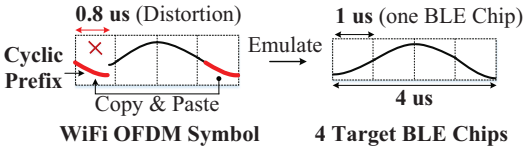


Figure 3: Previous OFDM Method Suffers from Severe Chip Errors Due to the Cyclic Prefix Restriction.

5 WIBEACON BROADCAST

WiBeacon takes an approach different from previous designs. We observe that besides OFDM, every WiFi AP also supports complementary code keying (CCK) modulation specified in a legacy WiFi protocol[1], which is mandatory for backward compatibility - even the latest WiFi 6 APs [3, 4, 10] must be able to serve legacy WiFi clients.

In this section, we propose "CCK method". Interestingly, our CCK method suffers from a comparable amount of signal imperfections as previous OFDM method. Yet, it manages to produce no chip errors and is fully compatible with unmodified mobile devices. We first present its basic idea in Section §5.1. Then we analyze and tackle two critical hardware restrictions of CCK method in Section §5.2 and §5.3.

5.1 WiBeacon Broadcast via CCK

5.1.1 CCK Preliminary. Fig.4(a) demonstrates the conceptual diagram of CCK modulator. Every 8 bits are encoded into a *codeword* (denoted as C_i) containing 8 complex chips (i.e, $C_{i,0}, ...C_{i,7}$) according to a *codebook* defined in [26]. The chips are modulated by Quadrature Phase Shift Keying (QPSK). As Fig.4(b) depicts, each QPSK chip $C_{i,j}$ takes one of the four values ($e^{j0}, e^{j\frac{\pi}{2}}, e^{j\pi}, e^{-j\frac{\pi}{2}}$) which correspond to 4 quadrature phases (0°,90°,180°,−90°). QPSK chips are transmitted sequentially in 11 MHz. Each QPSK chip takes $\frac{1}{11}\mu$s and it takes $\frac{8}{11}\mu$s in total to transmit a CCK codeword of 8 chips.

5.1.2 CCK Method: Basic Idea. To demonstrate the basic idea, we temporarily assume that CCK modulator does not have any hardware restrictions, i.e., it can generate arbitrary QPSK sequences

[1]Note that WiFi APs support multiple 802.11 protocols simultaneously (e.g., transmitting control frames such as beacons, probe requests in 802.11b for maximum reliability, while using 802.11g/n/ac for high-rate data communications). Hence, WiBeacon does not prevent APs from serving WiFi clients with high-rate 802.11 protocols.

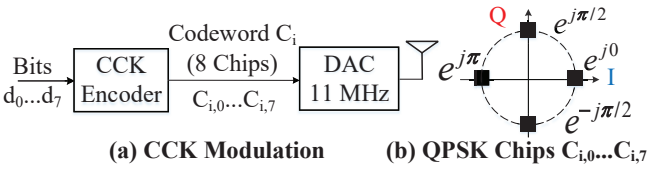


Figure 4: Complementary Code Key (CCK) Modulation Produces 11 MHz QPSK Chips w/ Quadrature Phases and Transmit in Sequence.

(assumption #1) at an arbitrary frequency (assumption #2). Note that both assumptions will be relaxed in Section §5.2 (assumption #1) and §5.3 (assumption #2).

Recall that BLE chips are modulated by phase shifts (i.e., phase changes) between samples, while each QPSK chip of CCK produces a quadrature phase at a specific sample. Therefore, by manipulating QPSK chips (i.e., phases) over time, we can create phase shifts that perfectly emulate BLE chips. In the upper and middle parts of Fig.5, we demonstrate an example of emulating a positive phase shift via QPSK chips. A BLE chip (denoted as $T_0 \rightarrow T_1$) is of 1 us while a QPSK chip (denoted as #1, ...#22) is of $\frac{1}{11}$ us duration. Thus, a positive BLE chip can be produced by a sequence of eleven consecutive e^{j0} (#1 − #11) followed by eleven consecutive $e^{j\frac{\pi}{2}}$ (#12 − #22). Vice versa for negative BLE chips.

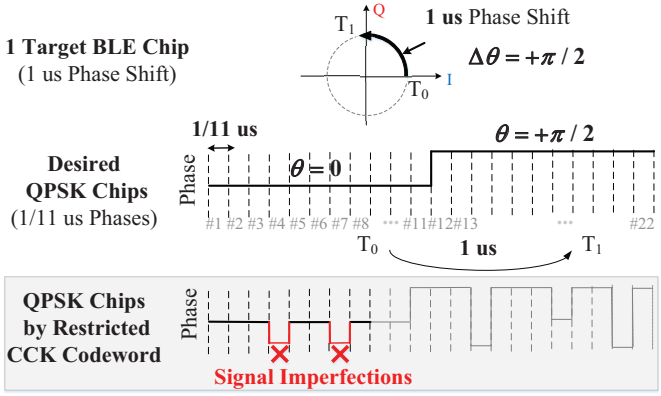
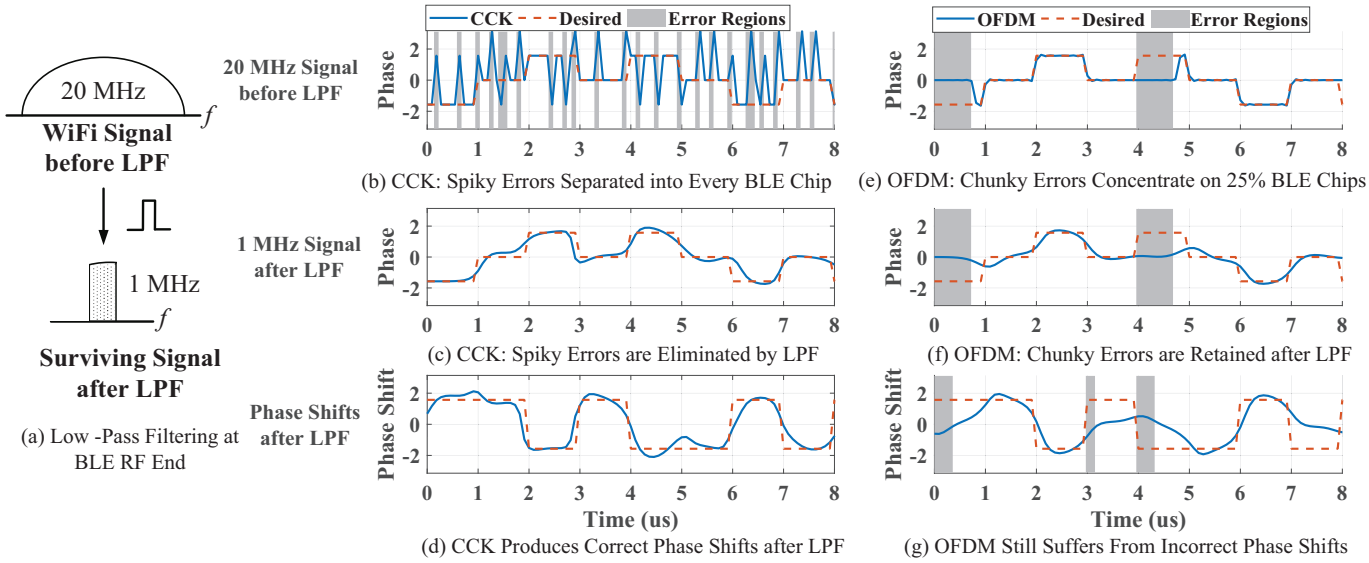


Figure 5: CCK Method: Target BLE Chip (top), Basic CCK Method (middle), and Signal Imperfections Caused by Restricted CCK Codewords (bottom).

5.2 Tackle Signal Imperfection

5.2.1 Codebook Restrictions. Unfortunately, the straightforward method in Section §5.1.2 cannot be directly applied. Every type of modulator in commodity wireless radios comes with *hardware restrictions* and CCK is not an exception. As discussed in Section §5.1.1, QPSK chips are created by CCK encoder in the form of codewords (i.e., a group of 8 QPSK chips). However, CCK *codebook* contains only 256 valid CCK codewords, which is far from our assumption of being able to produce arbitrary QPSK sequence (Since each chip takes one of the four possible quadrature phases, the total number of possible combination of 8 chips is as large as 4^8 = 65536).



(a) Low -Pass Filtering at BLE RF End

(b) CCK: Spiky Errors Separated into Every BLE Chip

(c) CCK: Spiky Errors are Eliminated by LPF

(d) CCK Produces Correct Phase Shifts after LPF

(e) OFDM: Chunky Errors Concentrate on 25% BLE Chips

(f) OFDM: Chunky Errors are Retained after LPF

(g) OFDM Still Suffers From Incorrect Phase Shifts

Figure 6: WiBeacon Exploits Low-Pass Filtering (a) of an BLE Receiver to Eliminate Signal Imperfections. The BLE Signal Produced via Our CCK Method (b) has Spiky and Separated Errors (Grey Regions in (b)). Spiky Errors are Smoothed out by Low-pass filtering (c) and thus Incurs Zero Chip Errors (d). In Contrast, Previous OFDM Method (e) produces Chunky and Concentrated Errors (Grey Regions in (e)) that are Retained after Low-pass Filtering (f) and Causes Severe Chip Errors (g).

In fact, the restriction of CCK codebook gives rise to even more severe signal imperfections than previous OFDM methods. Take the emulation of a positive phase shift depicted in Fig. 5 for example. The first 8 QPSK chips are desired to be eight consecutive e^{j0} (i.e., [1 1 1 1 1 1 1 1]). However, eight consecutive e^{j0} is not a valid codeword in the codebook according to [26], while a close approximation (depicted in the bottom of Fig. 5) is [1 1 1 -1 1 1 -1 1], which differs from the desired one by 2 QPSK chips. In general, at least 25% of signals (2 out of 8 QPSK chips) generated by CCK codewords are imperfect, which is more severe than previous OFDM methods (Recall that 0.8 μs cyclic prefix in 4 μs OFDM symbol only causes 0.8/4=20% imperfections).

5.2.2 The Opportunity of Imperfection Elimination. To eliminate imperfections of CCK signal without modifying mobile devices, we exploit a hidden opportunity - low-pass filtering (LPF), a standard procedure on BLE receivers. As Fig.6(a) depicts, a BLE receiver employs a low-pass filter at its front end for noise reduction. The filter cuts off 20 MHz WiFi signal to 1 MHz, which is equivalent to a *moving average* of the input signal in the time domain. Our key observation is that by elaborately choosing CCK codewords with unique error patterns, imperfections in CCK codewords can be smoothed out by the low pass filter [2] and thus incur zero chip error.

Fig.6 (b) demonstrates the opportunity by comparing the desired BLE signal (dashed line in red) with a CCK signal (solid line in blue). Although the total number of imperfections (grey areas) is large, the duration of each instance of error (i.e., an unmatched QPSK chip) is extremely short (i.e., $\frac{1}{11}us$). Therefore, if we select CCK codewords correctly (the method will be discussed in Section §5.2.3), error regions appear to be short *spikes* that are *separated* into every BLE chip. When the signal pass through a 1 MHz low pass filter, these *spiky* and *separated* imperfections are dramatically smoothed out by the moving average of LPF, as depicted in Fig.6(c). Consequently, the filtered CCK signal yields exactly the same signs of phase shifts as the desired signal in Fig.6(d).

In contrast, previous OFDM method suffers from a different error pattern and thus cannot benefit from low-pass filtering. As depicted in Fig.6(e), instances of errors caused by 0.8*us* cyclic prefix are *chunky* rectangles and *concentrated* only one of four BLE chips. These *chunky* and *concentrated* imperfections are too long to be smoothed out by moving average, as Fig.6(f) depicts. Therefore, imperfections are retained after LPF, which causes severe chip errors in Fig.6(g).

5.2.3 CCK Codeword Selection. The remaining question is how to find CCK codewords that can take advantage of low-pass filtering and produce correct BLE chips. According to our previous analysis, a good codeword can be obtained in two steps. First, as discussed in Section §4.1, BLE chips are modulated by phase shifts. Hence, a good CCK codeword is expected to produce phases that are as similar to desired ones as possible. Therefore, we start with computing the desired QPSK sequence (denoted as C^d) with basic CCK method in Section §5.1.2. Then, we iterate through the codebook to find the codeword C_i that is closest to the desired sequence C^d in the phase domain (Equation 1). The distance in the phase domain is the summation of phase differences between C_i and C^d at 8 chips (i,e., $C_{i,j}$ and C^d_j, $j = 0...7$).

$$\arg\max_i \sum_{j=0}^{7} |tan^{-1}C_{i,j} - tan^{-1}C^d_j| \quad (1)$$

[2]Note that a low-pass filter is pervasively implemented on commodity BLE radios [36] for noise reduction. Thus, we do need any modification of mobile devices, as proved by 150 different smartphone models in Section §9.5.

The first step commonly yields several codewords with the same closest distance. To figure out the optimal one from the subset, we use our insight in Section §5.2 that separated imperfections can be smoothed out by low-pass filtering. Thus, for a codeword, the more separated unmatched QPSK chips are, the higher chances they can be eliminated by low-pass filtering. As Equation 2 illustrates, we pick the codeword in the subset, such that the minimum interval between consecutive unmatched chips is maximized. $I_{i,k}$ is the position of k^{th} mismatched chip of C_i compared to the desire C^d.

$$\begin{aligned} &\arg\max_i \min I_{i,k+1} - I_{i,k} \\ &s.t. C_{i,I_{i,k}} \neq C^d_{I_{i,k}} \end{aligned} \tag{2}$$

5.3 Misalignment Compensation

So far, we assume that the center frequencies of WiFi and BLE are perfectly aligned. However, restricted by the limited number of WiFi channels, they have to be misaligned. For example, an 8 MHz misalignment exists between BLE channel 39 (2480 MHz) and its nearest WiFi channel (2472 MHz).

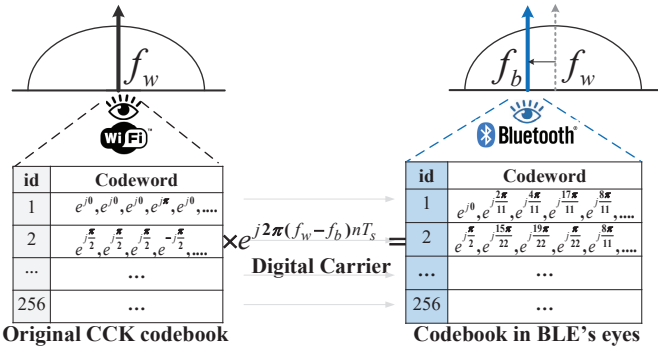


Figure 7: Codebook Adjustment Compensates Misalignments ($\Delta f = f_w - f_b = 1$ MHz in the example).

To address this issue, we propose *codebook adjustment*, a novel design to effectively compensate for up to 9 MHz misalignments. Our technical insight is that when emulating a BLE waveform with a misaligned center frequency, the WiFi transmitter essentially performs the emulation with an adjusted codebook from the BLE receiver's viewpoint. In specific, as the left part of Fig.7 depicts, the codebook in WiFi standards can be viewed as the codebook from WiFi's perspective where the receiver observes the WiFi signal in the same center frequency as WiFi transmitter (i,e., f_w). In contrast, the BLE receiver in the right part of the figure observes the WiFi signal from a different frequency point (i.e., f_b), leading to a different observation of each WiFi codeword.

Based on the key insight, WiBeacon adjusts the codebook to BLE's viewpoint before conducting CCK method. As Fig.7 demonstrates, if WiFi and BLE are operating in f_w and f_b respectively, we shift each codeword of the standard codebook in the frequency domain by $\Delta f = f_w - f_b$, which is done by the dot-product of each codeword with a digital carrier as Equation 3 illustrates, where T_s is the sample rate.

$$C_i^{adjusted} = C_i \cdot e^{2\pi(f_w - f_b)nT_s} \tag{3}$$

With the adjusted CCK codebook, we perform CCK method in Section §5.2.3 for each desired signal segment. Finally, calculated codewords are reverse engineered into data bits, i.e., WiFi payload. Note that this process is done entirely in the software, requiring no modification of WiFi hardware.

6 INTEGRATION WITH WIFI SERVICES

WiBeacon is integrated with WiFi services with a simple but effective method that is transparent to WiFi clients and applicable across AP devices of various vendors (Section §6.1). We also develop a dynamic WiBeacon scheduling algorithm to guarantee the reliability of iBeacon service while minimizing its impact on WiFi performance (Section §6.2).

6.1 Compatible Integration

The most critical task of the integration is to emulate frequency hopping function of iBeacon, while being transparent to WiFi clients. To achieve this, we propose a method that entirely uses standard features of 802.11 AP. As a result, it doesn't require any cooperation with WiFi clients. Neither does it cause stability issues (e.g., disconnections and packet loss). Note that we intentionally avoid using any hardware-specific features, so that our method can work across APs of different vendors (as proved by our implementation in Section §8). The method takes three steps:

Frequency hopping preparation: Before switching WiFi radio off the operating channel, we make several preparations to avoid packet loss. First, we disable the outgoing gate of data queue and buffer incoming data to prevent downlink WiFi packets from being erroneously transmitted. Second, to prevent uplink traffic from clients when the AP has switched away, we transmit cts-to-self - an 802.11 control frame that can silent the clients for a specific short period of time (< 33 ms) [43]. Upon receiving the frame, the clients set their NAV (network allocation vector) accordingly and would not attempt to send uplink traffic until NAV expires.

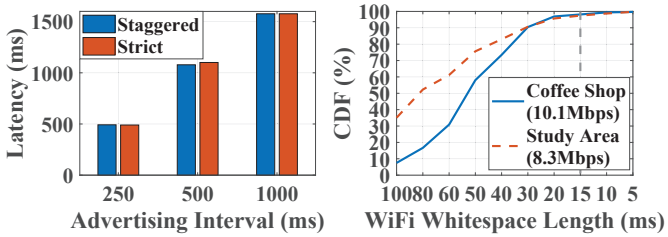
Frequency hopping: To switch WiFi radio to a different channel without disconnecting clients, we reuse *offchannel* function [14] of WiFi APs, which is an 802.11 standard feature design originally for APs to detect sources of interference or unauthorized ad-hoc networks. Offchannel function can switches the radio to another WiFi channel for a small amount of time (10-15 ms), while preserving entire states of the AP. Therefore, it avoids reset of the network and disconnecting clients. We use this opportunity to broadcast emulated iBeacon frames.

Frequency hopping completion: When the timer expires, we immediately switch WiFi radio back to the operating channel, so it starts to deliver buffered data to clients. At this time, WiFi clients can resume uplink data transmissions.

6.2 Dynamic WiBeacon Scheduling

WiBeacon is expected to guarantee the reliability of LBS, while minimizing its impact on the performance of WiFi service. To meet both goals, we carefully design a WiBeacon scheduler (depicted in Fig.9) based on two key observations. First, iBeacon broadcast is the connection-less, so it doesn't need to be strictly periodic and thus can be delayed to avoid interference with WiFi traffic. Second,

the Internet data traffic through WiFi is bursty in nature, leaving plenty of temporal whitespace for WiBeacon.



(a) Average Detection Latencies (b) WiFi Whitespace Distributions

Figure 8: Opportunities for Dynamic Scheduling.

To verify our observation, we conduct an empirical study where we compare the average time it takes for COTS smartphone to detect BLE beacon when the advertising frames are strictly periodic and staggered (delay by a random time between 0 and the advertising interval). Fig.8(a) depicts that in various advertising intervals the average detection latencies are not increased by the random delay. In addition, we analyze real-world wireless data traces of public APs in a coffee shop and study area during the peak hour. Both APs serve over 10 clients with 10.1 Mbps and 8.3 Mbps throughput. The distribution of lengths of contiguous whitespace in each 100 ms are plotted in Fig.8(b). Over 97.3% of the 100 ms periods have 15 ms or longer whitespace, which is sufficient for offchannel activity, which typically takes 10-15 ms.

Based on the observations, WiBeacon is scheduled to maintain a constant broadcast frequency while minimizing the delay of WiFi traffic. Specifically, we dynamically schedule WiBeacon broadcast as a periodic real-time task with a dynamic priority. As Fig.9 depicts, the period and deadline are assigned to be the iBeacon advertising interval, which is a parameter that users configure. To avoid interference with WiFi traffic, WiBeacon task has the lowest priority if not approaching the deadline. The scheduler monitors the data queue and opportunistically executes WiBeacon when the queue is empty. In this way, WiBeacon effectively utilizes the whitespace between bursty WiFi traffic.

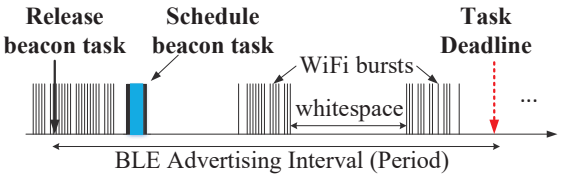


Figure 9: Explore WiFi Temporal Whitespaces

When WiBeacon task approaches the deadline, the scheduler increases its priority and starts the transmission of iBeacon broadcast frame. This introduces bounded delay (typically less than 15 ms) for WiFi data while preventing the starvation of BLE LBS. Note that such a situation is extremely rare in the real-world WiFi network. We analyze captured traces at multiple places with more than 10 users (e.g., coffee shop in Fig.8). When the advertising interval is set to 500 ms, the possibility is less than 0.01%.

7 LIMITATION AND DISCUSSION

7.1 Limitation

Applicable vs. Inapplicable Scenarios: WiFi APs are originally deployed for providing high-speed Internet access to mobile devices, which is a very different purpose from location services through BLE beacons. Thus, reusing existing WiFi APs for BLE LBS suffers from potential limitations. First, WiFi APs are normally deployed at a coarse granularity, e.g., one AP per room or point of interest. Hence, WiBeacon might not be suitable for indoor localization with high precision requirements (e.g., precisely tracking visitors in galleries). However, in these scenarios, WiBeacon could still offer location information with low maintenance cost through existing WiFi APs, while additional BLE beacons are required to further improve localization accuracy. Secondly, WiFi APs are commonly placed at places where Internet coverage is needed, while some applications might desire BLE beacons to be installed at different points of interest (e.g., the entry/exit to stores or certain aisles). This mismatch also requires additional deployments of BLE beacons to meet the demands of specific scenarios.

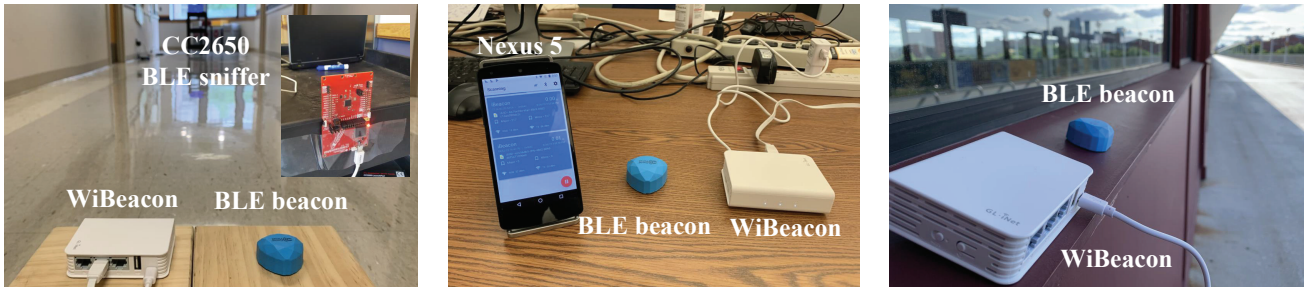
WiBeacon is most suitable to the emerging BLE beacon systems that are large-scale and geographically scattered. A concrete use case is the on-going city-wide location infrastructure deployments by Alibaba local service [27], in which every restaurant will install exactly one beacon while 2.5 million beacons are expected to be deployed nationwide. Since merchants have already installed at least one WiFi AP for providing free Internet service to customers, upgrading WiFi APs with WiBeacon is sufficient to support this use case. It can significantly reduce the deployment costs in such scenarios because the total number of beacons are extremely large, as well as the management costs because deployed beacons are scattered, and traditional beacon require extensive site visits for maintenance. Another killer application of WiBeacon is automatic check-in for mobile devices that provides accurate contract tracing with rapid deployment during the pandemic (e.g., COVID-19). Other applications of WiBeacon include location-aware to-do lists reminders and coupon delivery at retail stores [42].

Broadcast Channel Coverage: BLE standard defines three broadcast channels (i.e., Ch^b_{37}, Ch^b_{38}, and Ch^b_{39}). With codeword adjustment WiBeacon can effectively cover Ch^b_{38} and Ch^b_{39}, while Ch^b_{37} is too far away from any WiFi channel. As we will demonstrate in Section §9.4, the coverage of two broadcast channels is sufficient to provide reliable location-based services in real-world scenarios.

7.2 Discussion

Availability of Legacy WiFi: Every commodity WiFi AP must support 802.11b CCK modulation for backward compatibility (i.e., serving legacy clients). Thus, WiBeacon applies to both deployed and future APs. For example, CCK is available in the latest WiFi 6 APs [3, 4, 10]. Additionally, an AP can simultaneously operate with different 802.11 protocols, so WiBeacon does not force AP to work in 802.11b. Regular WiFi traffic can still use high-rate protocols, i.e., 802.11g/n/ac/ax, while we exclusively use CCK for WiBeacon.

Proximity Estimation Accuracy: By default, an AP has higher transmission power and hence a broader signal coverage. Note that a broader coverage does not mean less proximity estimation accuracy because proximity is not determined by whether the signal



(a) Hallway (b) Office (c) Outdoor Bridge

Figure 10: Evaluation Scenarios (Commodity WiFi AP and BLE Beacon at a Hallway, a Lab, and a Bridge).

of a beacon is received but how much signal is attenuated. Thus, even if devices receive WiBeacon signal at a farther distance, the larger value of signal attenuation measurements suggests it is still far away. Furthermore, as we will discuss in Section §9.2.4, we can tune down the power of WiBeacon while retaining LBS reliability.

WiBeacon Deployment: Installations of WiBeacon in deployed WiFi APs can be done in several ways. Retail stores, restaurant chains, universities, etc. often have remote administrative control of their private APs. These organizations have the incentives to upgrade their APs to serve their own mobile APPs. Additionally, public WiFi hotspots owned by Internet service providers (ISP) can also be upgraded remotely. For example, Xfinity runs over 18 million free public WiFi APs [60], so we can cooperate with them to deploy a high amount of WiBeacon for public or private usage.

The Generality of WiBeacon: In our paper, we only discuss WiFi to BLE communication. However, our key technical insight could be generalized and improve cross-technology communication from WiFi to other narrowband wireless techniques. Specifically, our critical observation that low-pass filter at narrowband front end can be exploited to eliminate communication errors is generally applicable to other CTC designs. For example, our experiment shows that CCK method also significantly increases the reliability of WiFi to ZigBee CTC from 50% to 99.9% compared to the previous OFDM method proposed in WEBee [48]. However, we gloss over the results in the paper because they are out of scope.

Security and Privacy: WiBeacon has also considered network security and public privacy on large-scale deployment. First, WiBeacon is protected by the sophisticated WiFi authentication mechanism, which is much stronger than Bluetooth beacons. In addition, WiBeacon will not harm public privacy because it only broadcasts identifiers without the ability to collect or listen to the Bluetooth packets in the air.

7.3 Future Works

In our future work, we will improve the practicality of WiBeacon from the following aspects. First, we plan to design a more efficient WiBeacon scheduler to further reduce the interference between WiBeacon and WiFi traffic. For example, we could apply data-driven WiFi traffic prediction for estimating future WiFi traffic and dynamically adjusting BLE beacon injection. Second, we will cooperate with WiFi device vendors and open-source communities to customize the firmware for WiBeacon. This allows us to reduce the overhead of switching WiFi channels in our frequency hopping implementation.

8 SYSTEM IMPLEMENTATION

Implementation for Large-scale Deployments: We implement WiBeacon on OpenWrt [12], the most popular open-source operating system for the commodity WiFi APs. Note that WiBeacon is not hardware-specific - it works with any WiFi drivers and chipsets since it only uses 802.11 standard features. To demonstrate this, we exert a lot of effort to adapt WiBeacon to commodity APs using WiFi radios from highly diversified chipset vendors. Table 1 lists part of the representative AP models we have tested.

Table 1: Representative WiFi Devices Tested.

Device	Chipset	Vendor	Driver
GL-AR750	QCA9531	Qualcomm	ath9k
Linksys-E2000	BCM4328	Broadcom	b43
Netgear-A6210	MT7612E	MediaTek	mt76

To facilitate future deployments on a large scale, we develop WiBeacon as OpenWrt .ipk packages, which can be downloaded and installed on APs in the same way as installing Android apps on smartphones. In addition to basic LBS functions, the package provides two features for better usability: a remote configuration interface for updating the beacon information (e.g., UUID) and a heartbeat service interface for remote monitoring.

9 PERFORMANCE EVALUATION

9.1 Evaluation Settings

WiBeacon is extensively evaluated in both lab environments and real-world commercial applications. As Fig.10 demonstrates, we test WiBeacon in several representative scenarios, including a hallway of the campus building, a lab office, and a 400-meter long bridge. In addition, we cooperate with a food delivery platform and conduct a pilot study at 20 restaurants in Shanghai over two weeks.

The performance of WiBeacon is evaluated with commodity WiFi APs (e.g., GL-AR750 [7]). We measure the low-level communication quality with CC2650 BLE sniffer [36], while compatibility with mobile devices and overall service performances are evaluated with 150 COTS smartphones of different models. The results are compared with a dedicated BLE beacon (Minew i8 [9] with nRF51822 chipset).

We start with detailed evaluations of two critical designs for service-level compatibility, i.e., the reliability of WiBeacon location broadcasts (§9.2) and the integration with WiFi service (§9.3). Then overall LBS performances (§9.4) are examined. Finally, we discuss our pilot study in the commercial LBS applications (§9.5). Without further explanations, both WiBeacon and BLE iBeacon transmit standard iBeacon frames with 36 bytes, with default transmission powers of 15 dBm and 4 dBm, respectively. In each experiment, 10000 frames are sent, and the statistical results are reported.

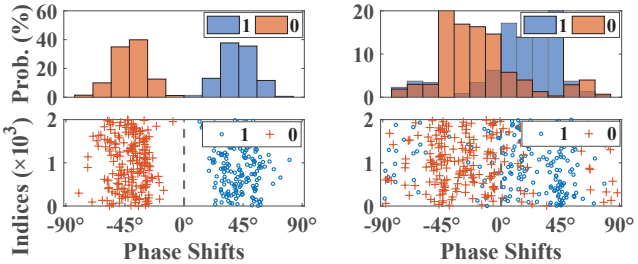


Figure 11: Chip Errors: CCK vs. OFDM.

9.2 WiBeacon Broadcast Reliability

9.2.1 Chip Accuracy: CCK vs. OFDM. Each bit of iBeacon broadcast is modulated into one chip (± phase shift). Due to the lack of redundancy, a single chip error causes frame corruption and incompatibility with mobile devices. To achieve service-level compatibility, we propose CCK method, which exploits low-pass filtering of a BLE receiver to eliminate chip errors. To demonstrate its effectiveness, we adopt CCK method and OFDM method proposed in [48] to generate the same iBeacon broadcast. Phase shifts after LPF are compared with the ground truth in Fig.11, where chip 1 is denoted as ∘ and chip 0 is plot as +. Chip errors occur when phase shifts cross the decision boundary depicted as the dotted line. As Fig.11(a) shows, CCK method is always able to produce correct phase shifts. In contrast, OFDM method in Fig.11(b) incurs around 19.25% chip errors. The result clearly shows that CCK method effectively overcomes hardware restrictions of WiFi transmitters.

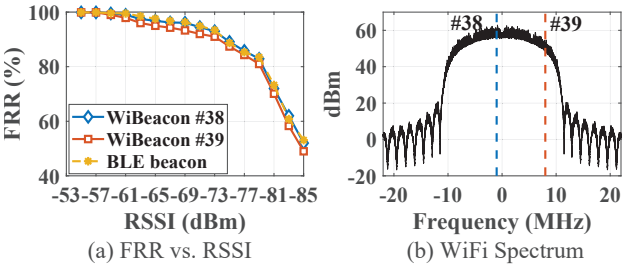


Figure 12: Frame Reception Ratio vs. RSSI.

9.2.2 Frame Reception Ratio: WiBeacon vs. BLE. We measure the frame reception ratio (FRR) of WiBeacon over varying received signal strengths and compare the results with the FRR of standard BLE beacons. The noise floor during our experiments is -95 dBm. The performances of WiBeacon on two BLE broadcast channels (Ch^b_{38} and Ch^b_{39}) are reported in the Fig.12 (a). The frame reception ratios of WiBeacon on both channels approximate standard BLE communication closely. It achieves > 95% FRR when the RSSI is above -65 dBm and manages to maintain 50% packet reception even when the signal is very weak (-85 dBm). This result demonstrates the accuracy of our CCK method, which is critical for reliable BLE location services. In addition, we observe that WiBeacon performs marginally better for Ch^b_{38} than Ch^b_{39}. This is because the two BLE channels are located at different positions of the WiFi spectrum as shown in Fig.12(b), which leads to a slightly different amount of emulation errors. However, as Fig.12 shows, the overall performances of the two channels are close to each other.

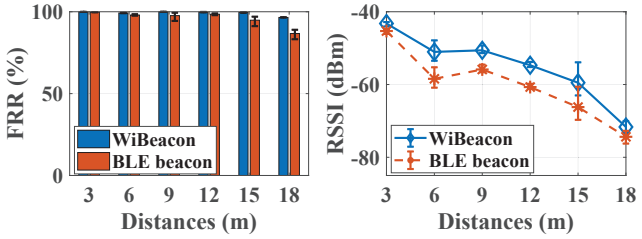


Figure 13: Indoor Performance: WiBeacon vs. BLE.

9.2.3 The Impact of Distances. To evaluate the communication distance of WiBeacon, we measure the frame reception ratio (FRR) of WiBeacon in both indoor and outdoor scenarios. In the experiment, GL-AR750 WiFi AP broadcasts at Ch^w_4 (2427MHz) using the default 15 dBm Tx power while the CC2650 BLE sniffer captures frames at BLE Ch^b_{38} (2426MHz). Fig.13(a) compares the FRR of WiBeacon with BLE beacon in the hallway. WiBeacon achieves higher accuracy (99.9% at 3 meters and > 97% within 18 meters) than BLE beacon because the stronger Tx power of WiFi transmitter provides a higher signal-to-noise ratio (SNR) at the BLE receiver. As demonstrated in Fig.13(b), the received signal strength of WiBeacon is 6 dB higher on average, which is smaller than the gap of transmission power (11 dB). This observation proves that BLE front end performs low-pass filtering that only passes through 1 MHz signal, making the effective bandwidth of WiBeacon 1 MHz. Consequently, we observe that WiBeacon and BLE beacon experience similar small scale fading due to multi-path effect (e.g., a sharp RSSI drop at 6 meters for both WiBeacon and BLE beacon in Fig.13(b)).

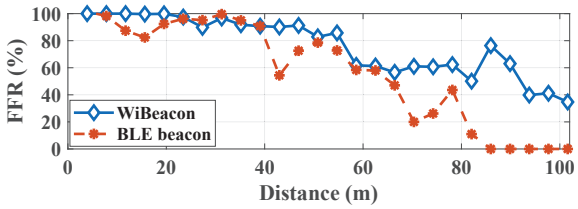
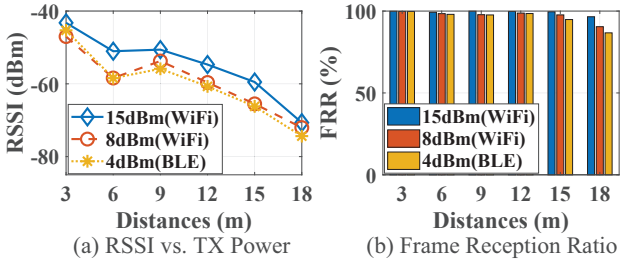


Figure 14: Outdoor FRR: WiBeacon vs. BLE.

In the outdoor scenario, the reception ratio of WiBeacon is even higher due to fewer wireless interferences. Fig.14 shows the FRR of WiBeacon and BLE beacon in varying distances on the bridge.

WiBeacon can deliver 99% frames at 20m and 90% at 40m. When the distance increases from 40 to 100 meters, the FRR slowly drops to 50%. The outdoor FRR of WiBeacon again outperforms BLE beacon due to the stronger transmission power of WiFi, which is extremely beneficial when WiBeaon needs to cover a very large outdoor area.



(a) RSSI vs. TX Power (b) Frame Reception Ratio

Figure 15: Impact of WiFi Transmission Power.

9.2.4 Impact of Tx Power. By default, WiFi device transmits with significantly higher transmission power than BLE devices. For a fair comparison, we also evaluate the performance of WiBeacon when the power is intentionally tuned low. Fig.15(a) shows that when the power is reduced to 8 dBm, RSSI of WiBeacon becomes comparable with standard BLE beacon. With lower power, WiBeacon is still reliable. As Fig.15(b) depicts, WiBeacon correctly delivers 90% BLE frames at 18 meters. Since a WiFi AP can adjust the transmission power for each frame, we can reduce the power of WiBeacon frames to produce fewer wireless interferences.

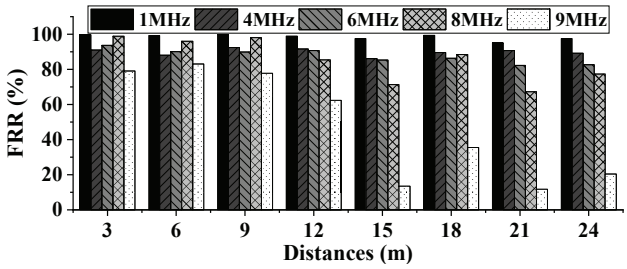
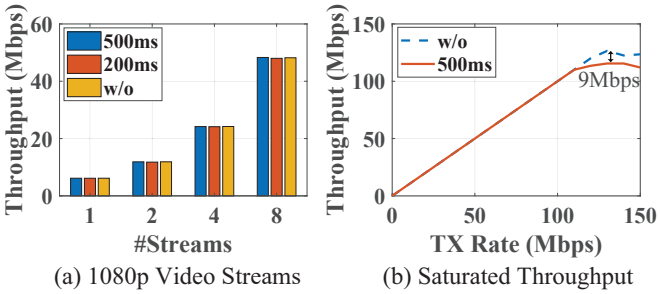


Figure 16: Frame Reception vs. Misalignments.

9.2.5 Impact of Center Frequency Misalignments. To overcome the restriction of WiFi channels, codebook adjustment is proposed for WiFi to produce iBeacon broadcast while having extremely large center frequency misalignments. Fig.16 shows that with an 8 MHz misalignment, WiBeacon achieves 90% FRR at 12 meters while up to 9 MHz can be tolerated. The capability allows WiBeacon to cover two BLE advertising channels, i.e., Ch_{38}^{b} (2426MHz) with Ch_{5}^{w} (2427MHz) and Ch_{39}^{b} (2480 MHz) with Ch_{13}^{w} (2472 MHz).

9.3 Integration with WiFi Services

9.3.1 Transparency to WiFi Service. To evaluate the transparency of WiBeacon to WiFi clients, we conduct experiments with one AP and eight clients and compare WiFi performance with and without WiBeacon. For the experiments to be repeatable, we capture packets of a 1080p YouTube video and replay them with Tcprelay[13].

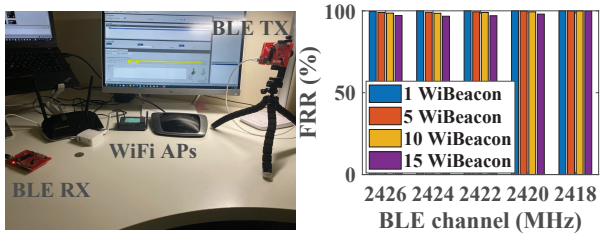


(a) 1080p Video Streams (b) Saturated Throughput

Figure 17: Transparency to WiFi Traffic.

Fig.17(a) presents the aggregated throughput under different advertising intervals and with a varying number of clients. In all cases, WiBeacon does not introduce noticeable throughput degradation.

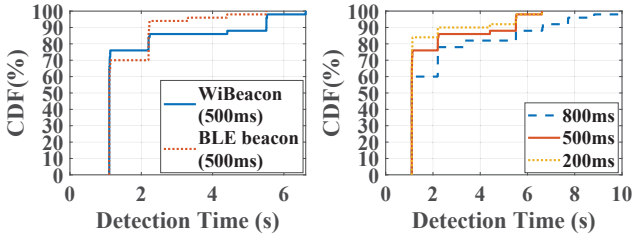
9.3.2 Saturated Throughput. We also examine the overhead when WiFi traffic is further pushed to the extreme. As Fig. 17(b) depicts, WiBeacon does not affect the throughput when the traffic is within 88%(110/125) of the saturated throughput for GL-AR750. When WiFi traffic is fully saturated, WiBeacon introduces a bounded overhead of 9 Mbps. This result coincides with our theoretical analysis. Since each frequency hopping takes 15 ms and WiBeacon performs two frequency hopping in one advertising interval (500 ms), the theoretical overhead is $(15 \times 2ms/500ms) \times 125 = 7.5Mbps$. Note that a WiFi AP is rarely fully saturated. For example, the utilization ratio under 8 YouTube streams is around 40%.



(a) Settings (b) FRR vs. BLE channels

Figure 18: Transparency to Ambient BLE Devices.

9.3.3 Transparency to Ambient BLE Devices. To examine the coexistence of WiBeacon with ambient BLE devices, we conduct an experiment as depicted in Fig.18(a). In the experiment, a various number of WiBeacon enabled AP broadcasts at Ch_{4}^{w} (2427 MHz) with a 500 ms beacon internal. We measure the frame reception ratios (FRR) of BLE communications at five BLE channels (from 2418 MHz to 2426 MHz). The transmission power of BLE transmitter is -15 dBm and the received signal strength is -55 dBm. As Fig.18(b) shows, the FRR is 99.99% with one AP while the reception ratio is above 97% when the number of APs are increased to fifteen. The result demonstrates that WiBeacon introduces moderate interference to BLE devices, thanks to listen-before-talk conducted by APs to effectively avoid collision with other wireless devices. In addition, the BLE channel that is further away from the center frequency of WiFi APs suffers less frame reception errors. For example, the impact of WiBeacon to 2418 MHz channel is almost negligible.

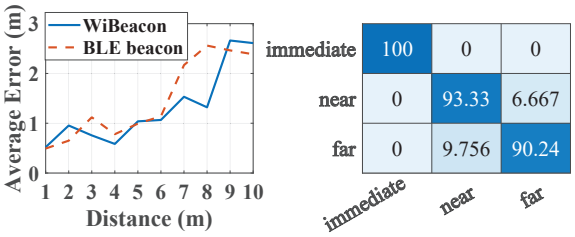


(a) WiBeacon vs. BLE Beacon (b) Varying Advertising Intervals

Figure 19: Distribution of Detection Interval.

9.4 Overall Service Performance

9.4.1 Beacon Detection. This section evaluates the detection latency of WiBeacon versus BLE beacon on commodity smartphones. It is noteworthy that to demonstrate the overall performance, WiBeacon only broadcasts at 2 BLE channels that WiFi signal covers (i.e., Ch^b_{38}, Ch^b_{39}), while a BLE beacon uses 3 BLE channels. Timestamps of detections are recorded on smartphones. Fig.19 depicts the distribution of the intervals between successive beacon detections of Nexus 5 at the distance of 10 meters, which are indicators of the latency. As Fig.19(a) depicts, when the advertising interval is 500ms, the latency of WiBeacon approximates BLE beacon tightly with a maximum of 6.5 seconds. The small margin exists because the extra Ch^b_{37} used by the BLE beacon slightly increases the detection probability. Furthermore, we evaluate the impact of advertising intervals. As the Fig.19(b) shows, more aggressive broadcasts (i.e., 200ms) further reduce the average latency while 500ms can achieve the best tradeoff between performance and overhead. Overall, the performance of WiBeacon is sufficient to serve popular LBS applications (e.g., check-ins and coupon delivery).



(a) Accuracy vs. Distance (b) Proximity Levels

Figure 20: Proximity Estimation Accuracy.

9.4.2 Proximity Estimation Accuracy. We use Android Beacon Library [1] to measure the proximity estimation accuracy of WiBeacon. The library estimates the distance by calculating signal attenuations (i.e., the ratio between the transmission power of a beacon and the received signal strength (RSSI) of smartphones). As Fig.20(a) depicts, WiBeacon demonstrates comparable accuracy with BLE beacon. The average error is within 1 meter when the distance is less than 5 meters. Both errors increase dramatically when the distance is large than 6 meters due to the uncertainty of RSSI. Apple categorizes the proximity into three coarse-grained ranges, i.e., immediate (< 0.5 meter), near (0.5-3 meters) and far (> 3 meters). We evaluate WiBeacon with the criteria. Fig.20(b) shows it distinguishes "immediate" from "near" with 100% accuracy and "near" from "far" with > 90% accuracy.

In summary, overall performances of WiBeacon are comparable to dedicated beacons, while it is superior to the dedicated beacon in deployment costs and management burdens.

9.5 Pilot Study

9.5.1 Real-world LBS Application. To demonstrate practicality of WiBeacon, we are cooperating with one of the largest Chinese food delivery companies and testing it in a real food delivery application. Specifically, this company recruits 100000+ meal couriers in Shanghai to deliver food from restaurants to customers. They need to deploy BLE beacons at 50000+ restaurants and use locations of couriers for automatic check-ins and new order dispatches. Since almost every restaurant has installed WiFi APs to provide customers free Internet service, WiBeacon can significantly cut down the deployment and maintenance cost.

Note that although couriers' smartphones have WiFi radios, extending WiFi APs for BLE LBS is still necessary due to two practical issues. First, the smartphone is required to continuously scan for reliable check-ins and real-time dispatches. A continuous WiFi scan quickly drains the battery, whereas BLE only consumes less than 2% extra power according to our measurement. Second, the service must be compatible with any potential smartphone of couriers. However, WiFi LBS does not work with any IOS smartphones because the check-in App on IOS cannot access scanned WiFi lists.

9.5.2 Pilot Study Settings. We remotely upgrade the APs in 20 restaurants near JinTie mall in Shanghai (depicted in Fig.21(a)), which then broadcast beacon identifiers to the meal couriers' smartphones. During this two-week pilot study, our WiBeacon system provides location-based services for 697 couriers, while helping 1780 orders in total. Note that the restaurants' environment can be highly complex with moving humans and obstacles, as demonstrated in Fig.21(a). During this deployment, the pre-built Openwrt packages are downloaded and installed on the APs. Then the system administrator configures the beacon identifiers via the remote access. This quick deployment demonstrates our unique benefits of zero additional hardware as well as the possibility of remote configuration/management.

After the quick deployment, these WiFi APs broadcast BLE identifiers at BLE channel 38 and 39 with 15 dBm transmission power, following an interval of 500ms. Upon receiving the packets from WiBeacon, the couriers' smartphones record the UUID, reception time, RSSI, the model of their smartphones, etc. To test our system, we compare it against a pre-installed BLE beacon (marked in Fig.21(a)) that is placed at the same spots with the 4 dBm Tx power. We evaluate the following aspects: device compatibility, signal strength, and LBS accuracy. Our observations are as follows.

• **Device Compatibility:** The diversified smartphone models used by the meal couriers provide us a comprehensive validation of WiBeacon in different smartphone models. As Fig.21(b) depicts, 150 types of smartphone models from 11 manufacturers (e.g., Apple, Huawei, and Samsung) have been proved compatible with WiBeacon, demonstrating that WiBeacon's design is generally applicable.

• **Signal Strength:** We analyze RSSI records of both WiBeacon and BLE beacons collected from couriers' smartphones. The results of 8 restaurants are demonstrated in Fig. 21(c). We observe that different from the results in the lab (Fig.13 (b)), WiBeacon does not

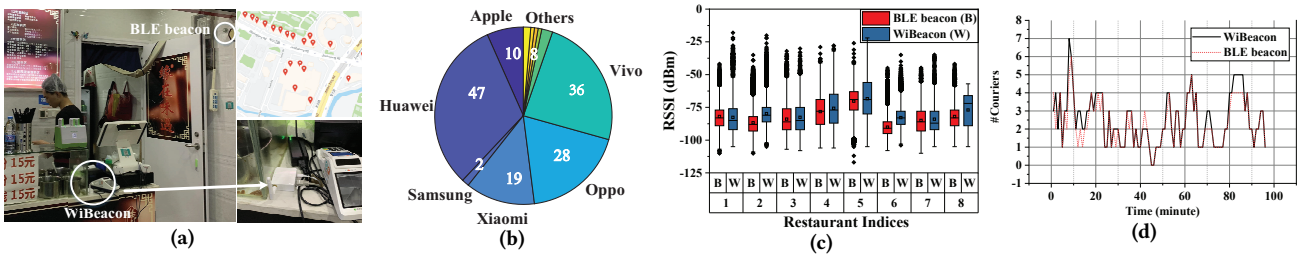


Figure 21: Pilot Study. (a) WiBeacon Deployments in 20 Restaurants. (b) 150 Smartphone Models Served by WiBeacon. (c) RSSI: WiBeacon vs. BLE Beacon. (d) WiBeacon is as Accurate as BLE Beacon in Courier Detections.

always have a better signal strength than BLE beacon. In fact, both RSSI vary significantly across different restaurants. This is because in complex real-world deployment, signal strengths are affected by many factors (e.g., the installation location of APs and beacons, the moving trajectory of the couriers, and smartphone models). Despite the huge RSSI fluctuation, the RSSI of WiBeacon and BLE beacons have similar trends (e.g., a similar median value and distribution pattern), demonstrating that WiBeacon is as reliable as dedicated BLE beacons.

•**LBS Accuracy:** We evaluate the location-based service performance of WiBeacon by measuring detections of couriers' arrival. The traces of BLE beacon in the same restaurant and order information from the food delivery platform are used as ground truth. During the two-week pilot study, WiBeacon detected every arrival events that are recorded by BLE beacons. Our interesting observation is during peak hours of the first weekend, WiBeacon detected two more arrival events than BLE beacons. A close look at traces reveals a unique benefit that was not observed in the lab - WiBeacon is more robust against WiFi interferences than BLE beacon because other WiFi devices naturally back off. Figure 21(d) visualizes the number of detected meal couriers by WiBeacon and BLE beacons in the same restaurant during peak hours (from 11 am to 1 pm). It is clear that the number of couriers that hear WiBeacon and BLE beacon is approximately the same throughout the time, suggesting WiBeacon's accuracy.

This two-week large-scale pilot study across 697 couriers and 150 smartphones proves that WiBeacon can provide reliable location services to real LBS application while significantly reducing the hardware cost and maintenance complexity. Based on this success, we are planning to deploy WiBeacon on a large scale for the further stress test.

9.6 RELATED WORKS

9.6.1 Location-based Service. Wireless localization has been extensively studied for various wireless protocols (WiFi, satellite, LTE, RFID, Bluetooth)[45, 50, 51, 54, 58, 59, 61, 62]. For example, researchers push the accuracy of WiFi localization to the sub-centimeter level. However, these works mainly focus on localization of the homogeneous technology. In contrast, WiBeacon proposes the first cross-technology location service for heterogeneous protocols. BLE beacon technology also gained significant attentions recently from both industry[2, 8] and academia[15, 21]. [27] deploys a city-wide BLE location service, which demonstrates the tremendous cost for deploying and maintaining large scale BLE LBS. To alleviate this issue, we propose the first low-cost BLE LBS solution using existing WiFi APs.

9.6.2 Cross-Technology Communication. Cross-technology communications (CTC) enables direct communication between heterogeneous wireless protocols. Early CTC works [20, 20, 23, 24, 30, 32, 33, 40, 44, 63, 65, 65, 66] design customized energy patterns to deliver messages and thus require significant modification at both wireless devices.

Recent advances in *physical-layer* CTC[18, 19, 22, 25, 31, 37, 39, 41, 48, 49, 64] directly emulate wireless signals of other protocols. The pioneering work (WEBee [48]) introduces signal emulation that enables WiFi radios to emulate ZigBee signal with OFDM modulator, thus achieving physical-layer compatibility with both WiFi and ZigBee. TwinBee [22], LongBee [49], and WIDE [31] further improve the reliability of WEBee. However, due to the inherent limitation of OFDM (e.g., cyclic prefix), these designs cannot avoid chip errors and thus still requires modifications on BLE receivers. WiBeacon is the first signal emulation technique using CCK modulation, which achieving service-level compatibility with both BLE LBS and native WiFi services. The direct communications between various IoT protocols (e.g., ZigBee, BLE and, LoRa) [39, 41, 46, 57] and between WiFi and LTE [19] are also studied, which mostly target physical-layer compatibility.

Integrations of packet-level CTC is studied in [34]. In contrast, WiBeacon is the first to rigorously integrate the existing LBS protocol on the service level.

9.6.3 Combo Devices. We notice that WiFi vendors (e.g., Cisco Meraki) recently integrate BLE beacon hardware into their high-end WiFi APs [16]. Although these new combo devices can provide the same service and similar benefit (e.g., low maintenance burden) as WiBeacon, it is impractical to replace all the deployed APs with new combo devices for large scale BLE LBS. In contrast, WiBeacon enables BLE LBS on already deployed WiFi APs via software upgrades, thus incurring no cost for deploying new hardware.

9.7 CONCLUSION

We propose WiBeacon, a cost-effective solution for large-scale BLE LBS. Our evaluation in the real-world application demonstrates that WiBeacon provides reliable BLE location services while incurring zero hardware cost and low management complexity.

10 ACKNOWLEDGE

This work was supported by the NSF CNS-1525235, NSF CNS-1718456, NSF CNS-1717059, CITYU APRC 9610491 and SRG ISTD 2020159. We sincerely thank our anonymous shepherd and anonymous reviewers for their valuable comments and feedback.

REFERENCES

[1] Android Beacon Library. https://altbeacon.github.io/android-beacon-library/).
[2] Apple. iBeacon. https://developer.apple.com/ibeacon/.
[3] ASUS GT-AX11000. http://en.techinfodepot.shoutwiki.com/wiki/ASUS_GT-AX11000.
[4] Buffalo WXR-5950AX12. http://en.techinfodepot.shoutwiki.com/wiki/Buffalo_WXR-5950AX12).
[5] Challenges between Mobile vs Beacon-Based Contact Tracing. https://kontakt.io/blog/challenges-between-mobile-vs-beacon-based-contact-tracing/.
[6] Gimbal Proximity Beacons. https://gimbal.com/beacons/.
[7] GL-AR750. https://www.gl-inet.com/products/gl-ar750/.
[8] Google. Project Eddystone. https://developers.google.com/beacons/.
[9] i8 Beacon. https://www.minew.com/bluetooth-beacons/i8-diamond-beacon.html.
[10] Netgear RAX200. http://en.techinfodepot.shoutwiki.com/wiki/Netgear_RAX200_(Nighthawk_Tri-Band_AX12).
[11] WiBeacon Github Repository. https://github.com/liux4189/WiBeacon.git.
[12] OpenWrt Project. https://openwrt.org/, 2019.
[13] Tcpreplay. https://tcpreplay.appneta.com/, 2019.
[14] WiFi Alliance. What is off-channel scanning for Wi-Fi access points (APs)? https://www.wi-fi.org/knowledge-center/faq/what-is-off-channel-scanning-for-wi-fi-access-points-aps, 2019.
[15] Roshan Ayyalasomayajula, Deepak Vasisht, and Dinesh Bharadia. Bloc: Csi-based accurate localization for ble tags. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*, pages 126–138. ACM, 2018.
[16] G. Bentinck. The benefits of integrated access point beacon technology. https://meraki.cisco.com/blog/2015/01/the-benefits-of-integrated-access-point-beacon-technology/, 2015.
[17] S. Bernstein. The realities of installing iBeacon to scale. https://www.brooklynmuseum.org/community/blogosphere/2015/02/04/the-realities-of-installing-ibeacon-to-scale/, 2015.
[18] Yoon Chae, Shuai Wang, and Song Min Kim. Exploiting wifi guard band for safeguarded zigbee. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 172–184. ACM, 2018.
[19] Eugene Chai, Karthik Sundaresan, Mohammad A Khojastepour, and Sampath Rangarajan. Lte in unlicensed spectrum: Are we there yet? In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 135–148, 2016.
[20] Kameswari Chebrolu and Ashutosh Dhekne. Esense: communication through energy sensing. In *Proceedings of ACM MobiCom 2009*, 2009.
[21] Dongyao Chen, Kang G Shin, Yurong Jiang, and Kyu-Han Kim. Locating and tracking ble beacons with smartphones. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 263–275. ACM, 2017.
[22] Yongrui Chen, Zhijun Li, and Tian He. Twinbee: Reliable physical-layer cross-technology communication with symbol-level coding. In *Proceedings of IEEE INFOCOM 2018*, 2018.
[23] Zicheng Chi, Zhichuan Huang, Yao Yao, Tiantian Xie, Hongyu Sun, and Ting Zhu. Emf: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous iot devices. In *Proceedings of IEEE INFOCOM 2017*.
[24] Zicheng Chi, Yan Li, Hongyu Sun, Yao Yao, Zheng Lu, and Ting Zhu. B2w2: N-way concurrent communication for iot devices. In *Proceedings of ACM Sensys 2016*, 2016.
[25] Zicheng Chi, Yan Li, Yao Yao, and Ting Zhu. Pmc: Parallel multi-protocol communication to heterogeneous iot radios within a single wifi channel. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, pages 1–10. IEEE, 2017.
[26] IEEE Computer Society LAN/MAN Standards Committee et al. Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11ˆ*, 2007.
[27] Yi Ding, Liu Ling, Yang Yu, Liu Yunhuai, He Tian, and Zhang Desheng. From conception to retirement: a lifetime story of a 3-year-old operational wireless beacon system in the wild. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*.
[28] Forrester. Pick The Right Location Technologies To Support Customer Experience And Operational Initiatives. https://www.forrester.com/report/Pick+The+Right+Location+Technologies+To+Support+Customer+Experience+And+Operational+Initiatives/-/E-RES120001, 2016.
[29] Apple Developer Forums. Is there any API available for iOS to scan WiFi networks. https://forums.developer.apple.com/thread/39204.
[30] Piotr Gawlowicz, Anatolij Zubow, and Adam Wolisz. Enabling cross-technology communication between lte unlicensed and wifi. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 144–152. IEEE, 2018.
[31] Xiuzhen Guo, Yuan He, Jia Zhang, and Haotian Jiang. Wide: physical-level ctc via digital emulation. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 49–60. IEEE, 2019.
[32] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Liangcheng Yu, and Omprakash Gnawali. Zigfi: Harnessing channel state information for cross-technology communication. In *Proceedings of IEEE INFOCOM 2018*, 2018.
[33] Xiuzhen Guo, Xiaolong Zheng, and Yuan He. Wizig: Cross-technology energy communication over a noisy channel. In *Proceedings of IEEE INFOCOM 2017*.
[34] Rainer Hofmann, Carlo Alberto Boano, and Kay Römer. X-burst: Enabling multi-platform cross-technology communication between constrained iot devices. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2019.
[35] SPEC INDIA. Beacons At Airport: The Next Big Thing In The Airlines Industry. https://www.spec-india.com/blog/beacons-at-airport-the-next-big-thing-in-the-airlines-industry, 2019.
[36] Texas Instruments. CC2650 SimpleLink™ Multistandard Wireless MCU. http://www.ti.com/lit/ds/symlink/cc2650.pdf.
[37] Hassan Iqbal, Muhammad Hamad Alizai, Ihsan Ayyub Qazi, Olaf Landsiedel, and Zartash Afzal Uzmi. Scylla: interleaving multiple iot stacks on a single radio. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*, pages 346–352, 2018.
[38] H. Jiang and H. Jamba. To Solve China's Bike-Sharing Woes, Hangzhou and Shanghai Turn to Bluetooth and Geofencing. https://thecityfix.com/blog/solve-chinas-bike-sharing-woes-hangzhou-shanghai-turn-bluetooth-geofencing-hui-jiang-harshita-jamba/, 2019.
[39] Wenchao Jiang, Song Min Kim, Zhijun Li, and Tian He. Achieving receiver-side cross-technology communication with cross-decoding. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 639–652. ACM, 2018.
[40] Wenchao Jiang, Zhimeng Yin, Song Min Kim, and Tian He. Transparent cross-technology communication over data traffic. In *Proceedings of IEEE INFOCOM, 2017*, 2017.
[41] Wenchao Jiang, Zhimeng Yin, Ruofeng Liu, Zhijun Li, Song Min Kim, and Tian He. Bluebee: a 10,000 x faster cross-technology communication via phy emulation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, page 3. ACM, 2017.
[42] L. Johnson. Walgreens launches iBeacon pilot to bolster coupon personalization. https://www.retaildive.com/ex/mobilecommercedaily/walgreens-tests-ibeacon-to-bolster-mobile-coupon-personalization-awareness/, 2017.
[43] Kyoung-Hak Jung, Yuepeng Qi, Chansu Yu, and Young-Joo Suh. Energy efficient wifi tethering on a smartphone. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 1357–1365. IEEE, 2014.
[44] Song Min Kim and Tian He. Freebee: Cross-technology communication via free side-channel. In *In Mobicom*, 2015.
[45] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM computer communication review*, volume 45, pages 269–282. ACM, 2015.
[46] Lingang Li, Yongrui Chen, and Zhijun Li. Physical-layer cross-technology communication with narrow-band decoding. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pages 1–2. IEEE, 2019.
[47] Tianxing Li, Chuankai An, Ranveer Chandra, Andrew T Campbell, and Xia Zhou. Low-power pervasive wi-fi connectivity using wiscan. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 409–420, 2015.
[48] Zhijun Li and Tian He. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 2–14. ACM, 2017.
[49] Zhijun Li and Tian He. Longbee: Enabling long-range cross-technology communication. In *Proceedings of IEEE INFOCOM 2018*, 2018.
[50] Zhihong Luo, Qiping Zhang, Yunfei Ma, Manish Singh, and Fadel Adib. 3d backscatter localization for fine-grained robotics. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 765–782, 2019.
[51] Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Vlado Handziski, and Souvik Sen. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *Proceedings of the 14th international conference on information processing in sensor networks*, pages 178–189. ACM, 2015.
[52] Justin Manweiler and Romit Roy Choudhury. Avoiding the rush hours: Wifi energy management via traffic isolation. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 253–266. ACM, 2011.
[53] Meraki. WLAN Location Analytics. https://documentation.meraki.com/MR/Monitoring_and_Reporting/Location_Analytics.
[54] Rajalakshmi Nandakumar, Vikram Iyer, and Shyamnath Gollakota. 3d localization for sub-centimeter sized devices. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 108–119. ACM, 2018.
[55] M. Owen. NFL, NBA to tap into iPhone to boost fan experience in stadiums. https://www.businessinsider.com/nfl-ibeacons-in-new-york-for-super-

bowl-2014-2, 2019.

[56] E. Richman. Public Wi-Fi hotspots to grow to 432M globally by 2020. https://www.fiercewireless.com/wireless/public-wi-fi-hotspots-to-grow-to-432m-globally-by-2020-suggesting-possible-threat-to, 2016.

[57] Junyang Shi, Di Mu, and Mo Sha. Lorabee: Cross-technology communication from lora to zigbee via payload encoding. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pages 1–11. IEEE, 2019.

[58] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single wifi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178, 2016.

[59] Jue Wang and Dina Katabi. Dude, where's my card?: Rfid positioning that works with multipath and non-line of sight. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 51–62. ACM, 2013.

[60] Xfinity. What are Xfinity WiFi Hotspots and how do I connect? https://www.xfinity.com/mobile/support/article/xfinity-mobile-wifi-hotspots, 2019.

[61] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16. ACM, 2019.

[62] Jie Xiong and Kyle Jamieson. Arraytrack: A fine-grained indoor location system. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 71–84, 2013.

[63] Zhimeng Yin, Wenchao Jiang, Song Min Kim, and Tian He. C-morse: Cross-technology communication with transparent morse coding. In *Proceedings of IEEE INFOCOM 2017*, 2017.

[64] Zhimeng Yin, Zhijun Li, Song Min Kim, and Tian He. Explicit channel coordination via cross-technology communication. In *Proceedings of ACM MobiSys 2018*, 2018.

[65] Yifan Zhang and Qun Li. Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices. In *Proceedings of IEEE INFOCOM 2013*, 2013.

[66] Xiaolong Zheng, Yuan He, and Xiuzhen Guo. Stripcomm: Interference-resilient cross-technology communication in coexisting environments. In *Proceedings of IEEE INFOCOM 2018*, 2018.