B^2W^2: N-Way Concurrent Communication for IoT Devices

Zicheng Chi*, Yan Li*, Hongyu Sun, Yao Yao, Zheng Lu, and Ting Zhu
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
{zicheng1, liy1, hongyu, yaoyaoumbc, zheng2016, zt}@umbc.edu

ABSTRACT

The exponentially increasing number of internet of things (IoT) devices and the data generated by these devices introduces the spectrum crisis at the already crowded ISM 2.4 GHz band. To address this issue and enable more flexible and concurrent communications among IoT devices, we propose B^2W^2, a novel communication framework that enables N-way *concurrent* communication among WiFi and Bluetooth Low Energy (BLE) devices. Specifically, we demonstrate that it is possible to enable the BLE to WiFi cross-technology communication while supporting the concurrent BLE to BLE and WiFi to WiFi communications. We conducted extensive experiments under different real-world settings and results show that its throughput is more than 85X times higher than the most recently reported cross-technology communication system [22], which only supports one-way communication (i.e., broadcasting) at any specific time.

CCS Concepts

•**Networks** → **Sensor networks;**

Keywords

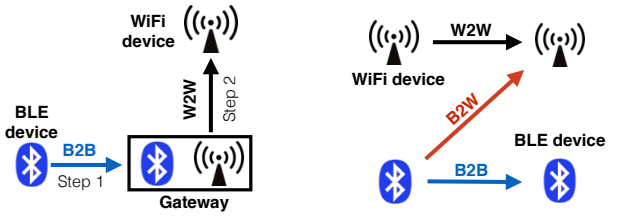
Wireless, Cross-technology, CPS, Internet of things (IoT)

1. INTRODUCTION

According to the research by Gartner, the number of internet of things (IoT) devices will grow exponentially to reach 26 billion by 2020 [1]. A lot of IoT devices use WiFi or Bluetooth Low Energy (BLE) that work at the industrial, scientific and medical (ISM) 2.4 GHz band. However, WiFi and BLE protocols are not compatible with each other at the physical layer. Therefore, a gateway (i.e., a bridge equipped with multiple radios) is normally used to connect these IoT devices. However, the gateway-based approach has two major issues: i) the additional cost to purchase the gateway hardware and ii) traffic overhead from a BLE device to a gateway, and then from the gateway to a WiFi device (shown in Figure 1(a)). When the number of IoT devices exponentially increases, the huge amount of data traffic generated by these IoT devices and the extra traffic introduced by gateways will significantly affect the wireless spectrum usage rate and eventually lead to spectrum crisis.

To address the above issue and enable more flexible and concurrent communication among IoT devices, we propose a new communication framework called B^2W^2, which can enable concurrent communications among IoT devices equipped with WiFi or BLE. The basic idea of B^2W^2 is to leverage the features of the overlapped narrower-band BLE channel and wider-band WiFi channel for embedding the message from a BLE into its overlapped WiFi sub-carriers. Figure 1(b) shows a simplified communication model of B^2W^2, which contains 3-way *concurrent* communications: i) WiFi to WiFi (W2W); ii) BLE to WiFi (B2W); and iii) BLE to BLE (B2B). By doing this, B^2W^2 eliminates additional hardware (i.e., gateways) and reduces extra traffic. Specifically, our major contributions are as follows:

- We have designed the first N-way *concurrent* cross-technology communication framework (B^2W^2) that enables BLE to WiFi communication while concurrently support the original WiFi to WiFi and BLE to BLE communications among multiple WiFi and BLE devices. Other state-of-the-art cross-technology communication methods can only support one-way communication at any specific time. For example, FreeBee [22] cannot conduct BLE to WiFi communication when there exists ongoing WiFi data traffic between two



(a) Gateway-based model, which needs two steps to enable the communication between a BLE device and a WiFi device.

(b) Simplified B^2W^2 model, which can support 3-way concurrent communications: W2W, B2W, and B2B.

Figure 1: Gateway-based model vs. B^2W^2 model

*Co-primary authors.

SenSys '16, November 14-16, 2016, Stanford, CA, USA

DOI: http://dx.doi.org/10.1145/2994551.2994561

WiFi devices.

- We extensively evaluated our design under one ideal setting (i.e., Faraday cage) and three different real-world settings (i.e., line-of-sight, non-line-of-sight, and multiple BLEs to WiFi communications). Results show that the throughput of B^2W^2 is more than 85X times higher than the most recently reported cross-technology protocol (i.e., FreeBee [22]) under the same setting.
- Our proposed method is generic and the basic concept (i.e., embedding narrower-band devices' message into broader-band devices' signal for concurrent transmission) i) can be applied to the coexistent IoT devices with regular Bluetooth radio and WiFi; and ii) has the potential to affect the next generation coexistent wireless networks design with extremely high spectrum utilization because our design utilizes existing WiFi and BLE traffic without introducing extra traffic.

2. CHALLENGES AND DESIGN OVERVIEW

In this section, we first discuss the design challenges and then introduce the overview of our design that overcomes these challenges.

2.1 Design Challenges

As described in the introduction section, N-way concurrent communication enables fascinating user experiences with multiple concurrent applications. However, in order to achieve N-way concurrent communication, we need to address the following key design challenges:

C1. How to send the message from BLE to WiFi with the concurrent transmissions of BLE to BLE and WiFi to WiFi? Since BLE and WiFi have totally different physical layer and bandwidth (i.e., 2 MHz for BLE and 20 MHz for WiFi), WiFi devices can not directly demodulate the BLE packets. Moreover, when two BLE devices communicate with each other, they have to follow the rules of FCC [2]: i) conducting frequency hopping over different channels with equal probability; ii) the hopping sequence will not repeat within 24 hours [2]; and iii) the period of BLE's transmission is fixed to be 625μs [2]. Therefore, it is very challenging to conduct concurrent transmissions of i) BLE to WiFi; ii) BLE to BLE; and iii) WiFi to WiFi. To address this challenge, we propose a novel modulation scheme (detailed in Section 3.1).

C2. How to accurately demodulate the message from BLE to WiFi under noisy environment? In order to demodulate the message from BLE to WiFi, we utilize the channel state information (CSI) values from the commercially available WiFi cards (e.g., Intel 5300 [15]). CSI is originally designed to capture the changes of the channel status. In our design, BLE devices communicate with WiFi devices by changing the CSI values detected by WiFi devices. However, CSI values can also be affected by the environment. The frequency hopping of BLE devices also introduces another issue because BLE devices' frequency may not always overlap with the WiFi device over time. In order to extract the BLE message under dynamically changing environment without modifying the hopping sequence of BLE devices, we need to carefully design the demodulation scheme at the WiFi side (detailed in Section 3.2).

C3. How to enable N-way concurrent communication among WiFi and BLE devices? As shown in Figure 1(b), our 3-way concurrent communication enables concurrent communication among 4 devices. With the exponentially increasing number of IoT devices, there is a pressing need of N-way concurrent communication among as many number of IoT devices as possible, which introduces the scalability issue. To address this issue, we introduce the novel design of N-way concurrent communication in Section 4.

2.2 Design Overview

In this section, we briefly describe our 3-way concurrent communication framework (shown in Figure 2), which is the basic design in our N-way concurrent communication framework. It can be extended to support N-way concurrent communication (detailed in Section 4). Our 3-way concurrent communication framework enables four IoT devices concurrently transmit the following three different pieces of message:

- M_{B2B}: the message from a BLE sender to a BLE receiver.
- M_{W2W}: the message from a WiFi sender to a WiFi receiver.
- M_{B2W}: the message from a BLE sender to a WiFi receiver.

Built on top of existing BLE and WiFi physical layers for handling the new type of message M_{B2W}, our 3-way concurrent communication framework contains two parts: i) modulation on the BLE device side, and ii) CSI extractor and demodulation on the WiFi device side.

On the BLE side, the modulation contains two components: the discrete amplitude and frequency-shift keying (DAFSK) converter (Section 3.1.2) and the symbol mapper (Section 3.1.3). By using the DAFSK converter, the M_{B2W} message from the upper layer of the BLE device will first be converted to a sequence of symbols. The value of these symbols indicate the transmission power levels of BLE radio. Then, the symbol mapper will map these symbols to a specific channel that the M_{B2B} message will be sent out. By doing this, we modulate the M_{B2W} message on top of the original M_{B2B} messages without affecting the hopping sequence between BLE device pairs.

On the WiFi receiver side, it contains CSI extractor (Section 3.2.1) and demodulator (Section 3.2.2). The CSI

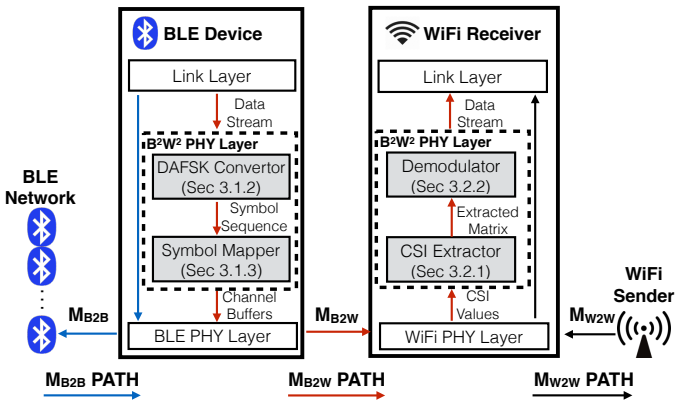


Figure 2: Overview of 3-way concurrent communication

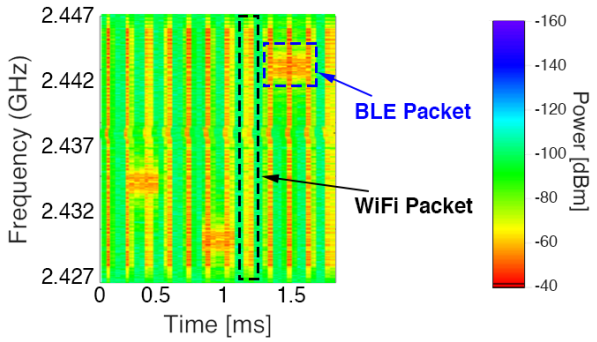


Figure 3: Power Spectral Density of WiFi and BLE

extractor extracts the embedded M_{B2W} message (in the symbol format) from the received WiFi packet by using the frequency domain correlation among adjacent CSI values. Based on the extracted symbols, the demodulator can reconstruct the embedded M_{B2W} message. Since we embed the M_{B2W} message in the CSI values of the WiFi packet, the regular WiFi packet M_{W2W} can also be decoded by the WiFi receiver.

The components in these two parts ensure the delivery of new BLE to WiFi message concurrently together with the original BLE to BLE and WiFi to WiFi communications.

3. 3-WAY CONCURRENT COMMUNICATION DESIGN

In this section, we first introduce the modulation on the BLE device side and then describe the CSI extractor and demodulation on the WiFi device side. At the end of this section, we also present how to establish the communication between the BLE sender and the WiFi receiver.

3.1 Modulation @ BLE

In this section, we introduce how to embed a M_{B2W} message into regular WiFi packets. We first introduce the design challenges and motivation, and then provide our detailed design.

3.1.1 Design Challenges and Motivations

To enable BLE to WiFi communication that is concurrent with B2B and W2W communications, there are two underlying challenges: i) BLE and WiFi have totally different physical layers (e.g., modulation schemes and bandwidth); and ii) BLE has to strictly follow the rules of FCC [2] to conduct frequency hopping over different channels with equal probability without repeating within 24 hours. Therefore, it is extremely difficult for the devices to communicate with each other while following WiFi and BLE standards. We have conducted extensive experiments and observed the following phenomena, which motivates the design of the modulation.

Observation 1: *The time duration of a normal WiFi packet is shorter than a BLE packet and WiFi's bandwidth is much larger than BLE. Therefore, multiple WiFi packets will collide with a single BLE packet.*

Observation 2: *Using CSI, a WiFi device can detect the fluctuation of a BLE's transmission power.*

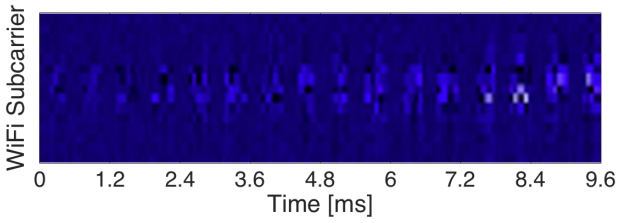


Figure 4: Recovered power level information based on the CSI value changes on the WiFi receiver side. The ligher the color, the higher the received BLE signal's power at the WiFi receiver.

In our experiments, an IEEE 802.11n WiFi device uses channel 6 (with 2.437 GHz central frequency) to send out WiFi packets and Texas Instruments CC 2650 BLE radio [3] sends out BLE packets using frequency hopping. Figure 3 shows the spectrum and time characteristics of WiFi and BLE signals in power spectral density. Signals in Figure 3 are sampled by using a National Instruments (NI) RF test bed in a WiFi channel. Figure 4 shows the detected CSI value changes at the WiFi subcarriers. In this experiment, the BLE device continuously changes its transmission power when sending out packets.

3.1.2 DAFSK Converter

Based on the above observations, it is possible that the WiFi device can detect the M_{B2W} message from BLE by monitoring changes of BLE's transmission power levels. Ideally, there are another two variables on the BLE side that we can utilize for modulation: packet length and packet interval. However, these two variables are not feasible, because according to the FCC regulatory specification of BLE, the period between two BLE transmissions is fixed to be 625μs [2]. So the only variable we can use for transmitting the message M_{B2W} is BLE's transmission power levels ($\{P_{tx,min}, \cdots, P_{tx,max}\}$).

Normally, BLE devices have eight to thirty-two levels of transmission power (e.g., the BLE radio CC2650 [3] from TI has thirteen transmission power levels). To improve the throughput, a naive modulation scheme is utilizing every transmission power level to indicate one state, which is represented by a sequence of bits. For example, a radio with eight power levels can use one power level to indicates three bits. The lowest transmission power level indicates "000", while the highest power level indicates "111".

However, there are two challenges. The first challenge is that transmission power can be attenuated and interfered with by other noise sources. The WiFi receiver may not be able to detect every transmission power level of the BLE devices. In other words, directly modulating based on transmission power levels will yield extremely high Bit Error Rate (BER) in dynamically changing environments. Therefore, Pulse-Amplitude Modulation (PAM) or Amplitude-Shift Keying (ASK) is not the best option. The second challenge is that BLE devices normally have limited computation and energy resources. It is very challenging to conduct complicated modulation schemes at the BLE device side.

To address these challenges and enable the reliable communication between BLE and WiFi under dynamically changing environments, we design the discrete amplitude and frequency-shift keying (DAFSK) converter to convert the data stream

from the upper layer of the BLE device into symbols that can be i) embedded into the regular BLE packets; and ii) accurately demodulated at the WiFi receiver side.

With multiple transmission power levels provided by the BLE radio, the basic idea of DAFSK is to form a basic sine wave by directly adjusting the adjacent BLE packets' transmission power levels. To provide the reliable B2W communication, we use the technique similar to Frequency-Shift Keying (FSK) by changing the frequency of the sine wave so that the data stream in the M_{B2W} message can be represented by the changes of the frequency. As shown in Figure 5, each discrete point on the sine wave is corresponding to the transmission power of a BLE packet. We can change the frequency of two discrete sine waves by changing their time duration. For example, we use a single sine wave with the total duration of 8 BLE packets and 4 BLE packets to represent bits "0" and "1", respectively. Therefore, the data stream "0011" from M_{B2W} message can be represented by a discrete sine wave with four symbols (shown in Figure 5).

3.1.3 Symbol Mapper

Based on the output symbols from the DAFSK converter, we can transmit two pieces of messages: i) the M_{B2W} is embedded into the regular B2B packets by setting the transmission power levels of every BLE packet and ii) the original M_{B2B} message is still transmitted by using tradional BLE packets. However, the frequency hopping mechanism of BLE devices introduces a major design challenge.

Frequency hopping is an interference avoidance mechanism adopted by BLE devices that periodically jump from one channel to another channel with fixed time interval 625 μs. At the beginning of each time slot, the BLE device quickly jumps to a new channel and immediately sends out a packet. Based on the FCC regulatory specifications [2], the hopping sequence should be pseudo-random and must not repeat within 24 hours. Therefore, the challenge is how to make sure that the WiFi device can correctly receive the M_{B2W} message without knowing the BLE device's randomized hopping sequences.

To address this challenge, we introduce the symbol mapper, which contains the following two steps:

Step 1: Serial to Parallel Conversion. Based on the number of channels that the BLE device will jump, we create the same number of first in first out (FIFO) buffers. Then we sequentially allocate the output symbols from the DAFSK converter into these buffers using serial to parallel conversion. As shown in Figure 6(a), if the BLE device only jumps between two channels, then the first element $P[0]$ is put in FIFO buffer 1 and the second element $P[1]$ is put in FIFO buffer 2.

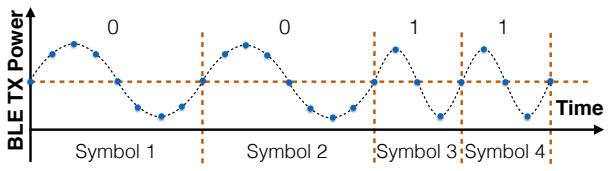
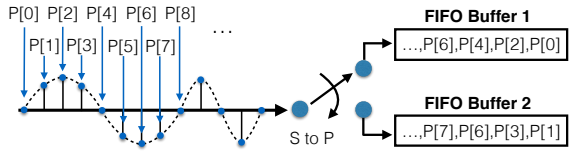
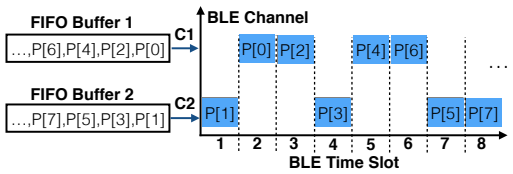


Figure 5: An example of converting "0011" from M_{B2W} message into symbols using the DAFSK converter. Each blue dot represents a BLE transmission power level to be tuned on an outgoing BLE packet.



(a) **Step 1:** Serial to Parallel Conversion



(b) **Step 2:** Mapping Tx. Power Levels to BLE Packets

Figure 6: Example of symbol mapping with 2 BLE Channels

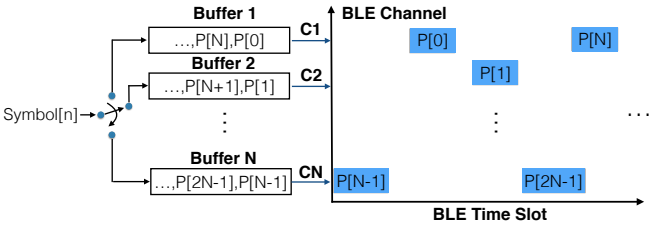


Figure 7: A General Architecture of Symbol Mapper

Step 2: Mapping Transmission Power Levels to BLE Packets. When a BLE sender (S_B) wants to send out the M_{B2B} message to a BLE receiver (R_B), based on the specification, S will conduct frequency hopping and send out the packet at the specific time in the selected channel. To enable the communication between BLE and WiFi (concurrently with B2B), S will be based on the selected channel to pick the first element ($P[i]$) in the corresponding FIFO buffer and tune its transmission power based on the value of $P[i]$. As shown in Figure 6(b), the BLE sender (S_B) needs to send a BLE packet to the BLE receiver (R_B). At the same time S wants to communicate with a WiFi receiver (R_W). In the first time slot, S_B jumps to channel $C2$, so S_B will go to buffer 2 to pick the first element (i.e., $P[1]$) and send out the BLE packet with the transmission power level equals to $P[1]$. In the second time slot, S_B jumps to channel $C1$, so S_B will go to buffer 1 to pick the first element (i.e., $P[0]$) and send out the BLE packet with the transmission power level equals to $P[0]$. By doing this, we do not need to change the hopping sequences of the BLE devices.

As described above, our symbol mapper method can support randomized hopping sequences and can be extended to support any number of BLE channels. Figure 7 illustrates the general architecture of our symbol mapper that can support arbitrary number (N) of BLE channels. Based on our observations (described in Section 3.1), during concurrent transmission, multiple WiFi packets will collide with BLE packets and a WiFi receiver can detect the fluctuation of a BLE's transmission power. Therefore, when a WiFi receiver detects the power level changes in the BLE channel by monitoring CSI, it will use the reverse way to reconstruct the

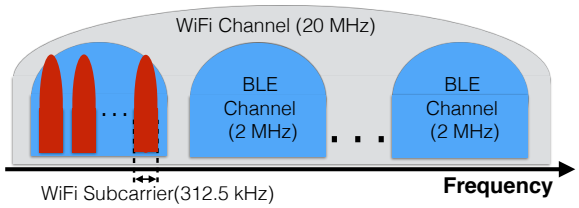


Figure 8: The relationship between a WiFi channel and BLE channels

symbol (detailed in Section 3.2.2).

3.2 CSI Extraction and Demodulation @ WiFi

In this section, we introduce a novel mechanism that allows the WiFi receiver concurrently recover the M_{B2W} and M_{W2W} messages from the concurrently transmitted BLE and WiFi packets. Specifically, we use the CSI extractor to extract the BLE packet's power level information, then use the demodulator to extract the M_{B2W} message. At the same time, the M_{W2W} message is still demodulated by using WiFi demodulator.

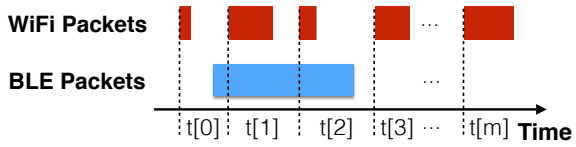
3.2.1 CSI Extractor Design

Channel State Information (CSI) is normally used by the WiFi system to measure the channel status from the sender to the receiver. Whenever the WiFi receiver receives a packet, it calculates the CSI values which includes the phase and magnitude attenuation caused by environmental changes at the subcarrier level. The CSI values can be accessed from commercially off-the-shelf WiFi cards (e.g., Intel 5300 [15]).

As shown in Figure 8, a typical WiFi channel (20MHz) is much wider than a BLE channel (2MHz). However, the WiFi system with Orthogonal Frequency Division Multiple (OFDM) divides the 20Mhz band into tens of subcarriers (e.g., 52 subcarriers in 802.11a/g/n/ac) and yields an equivalent 312.5KHz bandwith for each subcarrier. The bandwidth of a WiFi subcarrier is several times narrower than one BLE channel. Therefore, BLE packets will affect the CSI values of WiFi subcarriers. In our design, we want to use CSI to identify the BLE packet's power level. We introduce a novel CSI extractor on the WiFi receiver side to extract the CSI values that are affected by BLE's transmission. This module is one of the key component to enable 3-way concurrent communication and introducing zero extra traffic to existing WiFi and BLE infrastructures.

As shown in Figure 9(a), two WiFi packets were hit by a BLE packet. Therefore the the CSI values of these two WiFi packets (at time $t[1]$ and $t[2]$) will be affected. In the frequency domain, since one BLE channel is several times wider than the OFDM subcarrier (shown in Figure 8), one BLE channel will affect multiple subcarriers at once. Figure 9(b) shows the matrix of the WiFi receiver's CSI readings from time $t[0]$ to $t[m]$. In the matrix, the values in each column are the CSI readings at different time, and the values in each line are CSI readings of different subcarriers. For example, the $|CSI_n[m]|$ indicates the CSI reading of N-th subcarrier at time $t[m]$. Therefore, the BLE packet only affects the blue colored subcarriers from $i+1$ to $i+j$ at time $t[1]$ and $t[2]$ (see Figure 9(b)).

Now we introduce how to recover the extract the M_{B2W} message from the blue colored CSI readings which contains



(a) 1 BLE packet hits 2 WiFi packets

(b) CSI readings from subcarriers 1 to N after received the WiFi packets at different time

Figure 9: An example to demonstrate that some of the WiFi subcarriers' CSI values changed because of a BLE packet hits the corresponding WiFi packets.

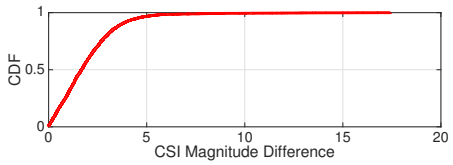


Figure 10: Empirical results to demonstrate the stability of CSI values in frequency domain: between two neighboring subcarriers in frequency domain, 80% of CSI magnitudes have the difference less than 3.

two parts: i) environmental noise and ii) M_{B2W} message. We first take a look at the adjacent CSI values in the matrix. As shown in Figure 9(b), the purple colored neighboring values in frequency domain are not affected by the BLE packet. We also conducted multiple experiments in real-world settings and observed the following phenomena:

Observation 3: *In the frequency domain, the neighboring subcarriers (e.g., subcarrier i and $i+1$) will have very similar CSI readings due to the similar wavelength and multipath effect (see Figure 10).*

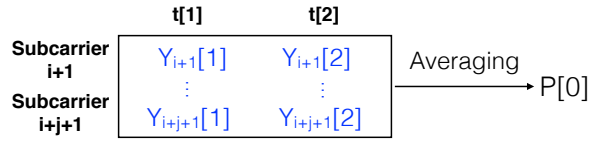
Based on the above observation, we can extract the information that the BLE packet embedded into the CSI values part. For example, the information that the BLE packet embedded into the blue colored value $|CSI_{i+1}[1]|$ part can be calculated using the following equation:

$$Y_{i+1}[1] = |CSI_{i+1}[1]| - |CSI_i[1]| \quad (1)$$

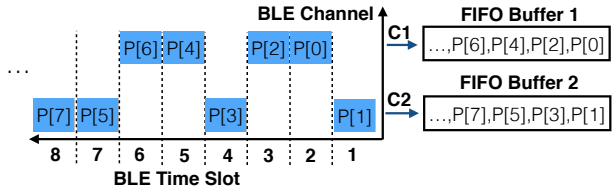
More generically, we can use the following equation:

$$Y_{i+1}[k] = |CSI_{i+1}[k]| - |CSI_i[k]| \quad (2)$$

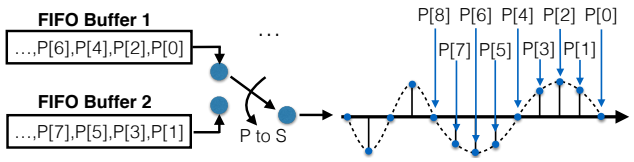
Where $Y_{i+1}[k]$ is just one element of the extracted matrix. We use the same way to get the whole extracted matrix $\mathbf{Y}$ which will be used as the input of the demodulator.



(a) **Step 1:** Matrix Averaging



(b) **Step 2:** Reconstructing Magnitude



(c) **Step 3:** Parallel to Serial Conversion

Figure 11: Example of symbol reconstructing with 2 BLE Channels

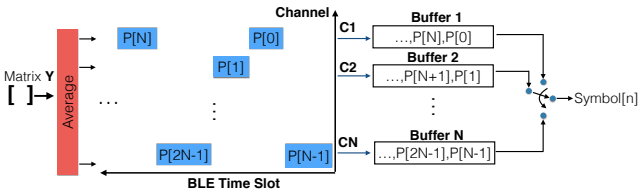


Figure 12: Architecture of symbol reconstructor

3.2.2 *Demodulator*

In this section, we introduce the demodulator at the WiFi receiver, which can recover the M_{B2W} message from matrix $\mathbf{Y}$. The demodulator contains two components: i) symbol reconstructor and ii) inverse DAFSK converter.

The symbol reconstructor at the WiFi receiver takes the extracted matrix $\mathbf{Y}$ from the CSI extractor (Section 3.2.1) as input and the output is the symbol sequence.

To compatible with the symbol mapper and address the challenge in Section 3.1.3, we introduce the symbol reconstructor on WiFi receiver, which contains the following three steps:

Step 1: Matrix Averaging. After fetching the subset (impacted by one BLE packet) of the extracted matrix $\mathbf{Y}$ from the CSI extractor, we average all the values to be one magnitude value which is proportional to the transmission power level of each BLE packet. As shown in Figure 11(a), all the blue colored CSI magnitude values are averaged to be one magnitude value $P[0]$.

Step 2: Reconstructing Magnitude. The averaged value pops out one by one at each BLE time slot and goes into different FIFO Buffers corresponding to its channel number. As shown in Figure 11(b), the first value $P[1]$ pops out at BLE time slot 1, since it is on the channel C2, the $P[1]$ goes into FIFO buffer 2. The second value $P[0]$ appears at BLE time slot 1 on channel C1 and goes into FIFO buffer 1.

Step 3: Parallel to Serial Conversion. The parallel to serial convertor flips among each channels to reorder the values from different channels into one symbol sequence. As shown in Figure 11(c), if there are only two channels, $P[0]$ and $P[1]$ from FIFO buffer 1 and FIFO buffer 2 are reordered to be a discrete sine wave.

As described above, our symbol reconstructor is symmetric with the symbol mapper (on BLE device) to deal with the randomly hopping issue and can be extended to support any number of BLE channels. Figure 12 illustrates the general architecture of the symbol reconstructor.

After the reconstructor, we have the complete symbol sequence and are able to recover the discrete sine wave by the inverse DAFSK converter. Then we can demodulate and obtain the data stream by measuring the frequency changing of the discrete sine wave which is just the inverse way of modulation (in Section 3.1.2).

3.3 Communication Establishment

In this section, we investigate the following questions: i) how does the WiFi receiver know the transmission of the M_{B2W} message from BLE device is about to start; and ii) how does the WiFi receiver differentiate the M_{B2W} message from normal BLE transmissions.

The BLE device takes the charge to initialize the communication. At the beginning, the BLE device sends a preamble that traverses all the transmission power levels (as a cosine wave) starting and ending with the maximum value $P_{tx,max}$ in order to inform the WiFi receiver.

Firstly, because the $P_{tx,max}$ has the most significant impact on the CSI values, the CSI extractor on the WiFi receiver (in Section 3.2.1) can immediately discover the BLE's transmission by performing a real-time calculation deriving from Equation 2. When the CSI extractor is in discover mode, we eliminate the component on from neighboring subcarriers because we do not know which subcarrier will be affected by BLE. And we apply it across all the subcarriers, the Equation 2 becomes to:

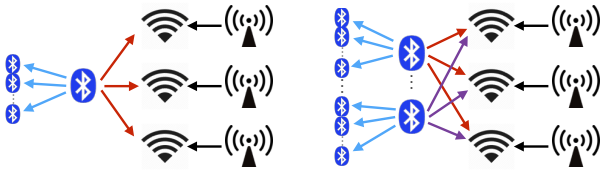
$$\vec{D}[m] = ||C\vec{S}I[m]|| - ||C\vec{S}I[m-1]|| \quad (3)$$

Where $||C\vec{S}I[m]||$ (e.g., the second column in Figure 9(b)) is the magnitude of CSI values from all subcarriers at time $t[m]$, $||C\vec{S}I[m-1]||$ (e.g., the first column in Figure 9(b)) is the previous values and $\vec{D}[m]$ is the difference. By tracking the $\vec{D}[m]$, if it is dramatically changed in a short time period, that means a BLE packet exists (e.g., the blue colored elements in Figure 9(b)).

Secondly, by continuously tracking the values in $\vec{D}[m]$ where $m = \{1, 2, 3, ...\}$, if the preset cosine wave pattern is observed (as sent on the BLE device), the WiFi receiver is able to differentiate the M_{B2W} message and normal BLE packets. This is because normal BLE packets will not continuously change their transmission power levels.

4. N-WAY CONCURRENT COMMUNICATION DESIGN

Building on top of the 3-way concurrent communication system described above, we generalize it to N-way concur-



(a) One BLE network to multiple WiFi device pairs (b) Multiple BLE networks to multiple WiFi device pairs

Figure 13: Scenarios of N-way concurrent communication

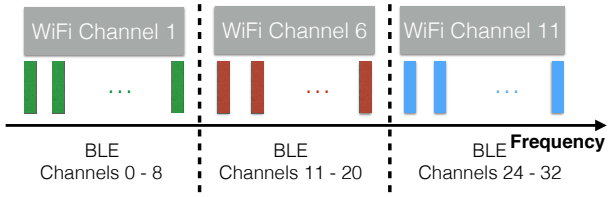


Figure 14: Overlaps of BLE channels with WiFi channels

rent communication. This scheme enables multiple BLE and WiFi devices concurrently communicate. We first introduce how to enable one BLE network communicate to multiple WiFi device pairs (One-to-Many) in Section4.1, then describe how to enable multiple BLE networks communicate to multiple WiFi device pairs (Many-to-Many) in Section 4.2. Figure 13 shows these two scenarios. Here, we use BLE networks instead of BLE node pairs. This is because one BLE master node can be connected to multiple BLE slave nodes to form a Piconet. The smallest BLE network is one BLE node pair, which includes one master node and one slave node.

4.1 One-to-Many

It is relatively easy to enable the concurrent communication between one BLE network and multiple WiFi device pairs. This is because the number of BLE channels can be up to 40 and each channel is 2 MHz. Therefore, the aggregated bandwidth of BLE can be up to 40 × 2 MHz = 80 MHz, which is much larger than WiFi's bandwidth. For example, the bandwidth of 802.11n is 20 MHz. Therefore, it is possible for the BLE device to i) adjust the number of channels to hop; and ii) embed different M_{B2W} messages to different WiFi channels. Thus M_{B2W} enabling concurrent communication between one BLE network and multiple WiFi device pairs. For example, Figure 14 shows some of the BLE channels are overlapped with WiFi channels 1, 6, and 11. The BLE device can concurrently embed 3 different pieces of M_{B2W} messages in WiFi channels 1 (i.e., BLE channels 0-8), 6 (i.e., BLE channels 11-20), and 11 (i.e., BLE channels 24-32), respectively.

At the WiFi receivers' side, since they are allocated on different channels, they can receive the different M_{B2W} messages from BLE by capturing the CSI values on their own channels (as introduced in Section 3.2). Therefore, there is no additional control needed on the WiFi side to enable one-to-many communication.

4.2 Many-to-Many

Ideally it should be easy to extend the one-to-many concurrent communication so that it can support many-to-many concurrent communication. However, the main challenge is how to recognize the sender of the BLE packets at the WiFi receivers' side. This is because of the following reasons: i) there is no explicit control message between BLE devices and WiFi devices because BLE and WiFi have totally different physical layer; ii) A WiFi receiver can not based on the hopping sequence to identify BLE devices because the hopping sequence of the BLE devices is random and will not repeat within approximately 24 hours; iii) in order to convey 1 bit information from a BLE device to a WiFi device, the BLE device needs to jump over multiple channels. Therefore, the BLE device can not put its node id or MAC address into the M_{B2W} message.

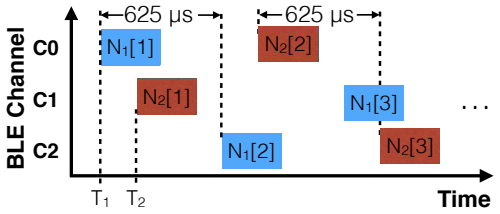


Figure 15: Two BLE networks Tx. to a WiFi receiver

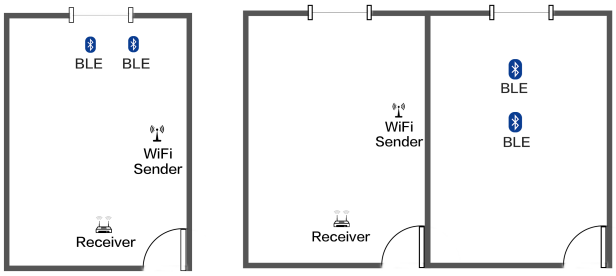
To address this challenge, we propose a simple solution that leverages the following unique features of the BLE networks [2]: i) to avoid the collision, different groups of BLE networks will have different hopping sequences and are not synchronized; and ii) the time duration for every frequency hopping is 625 μs and the transmission will start at the beginning of every frequency hopping. Therefore, the probability is extremely low when two different BLE networks start the transmission at exactly the same time in exactly the same channel.

Based on the above unique features, the WiFi devices can easily identify the BLE senders based on their unique starting time of each transmission. For the sake of clarity, let us consider the simplified scenario that contains two BLE networks (N_1 and N_2) and a WiFi receiver (shown in Figure 15). At time T_1, the WiFi receiver discovers the packet $N_1[1]$ from N_1 in channel C0. Then at time T_2, the WiFi receiver discovers the packet $N_2[1]$ from N_2 on channel C1. Since the two packets are in different BLE channel, they are distinguishable. After the first transmissions, both of the two BLE networks will jump to different channels. Based on the BLE specification, N_1 and N_2 will send their second packets exactly at $T_1 + 625\mu s$ and $T_2 + 625\mu s$, respectively. Therefore, the WiFi receiver can identify them even though it does not know the hopping sequences of N_1 and N_2.

5. EXPERIMENTAL EVALUATION

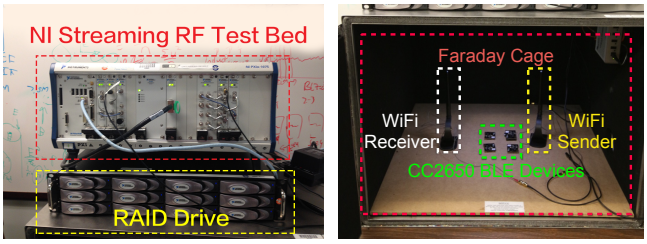
We have implemented a complete N-way B^2W^2 prototype using a National Instruments (NI) RF testbed and multiple Texas Instruments (TI) CC2650 BLE devices [3]. The modulation scheme we evaluated is DAFSK (shown in Figure 26(b)) as we described in Section 3.1. To compare with DAFSK modulation scheme, we also implemented PAM because the performance of PAM is approaching the theoretical limit in Faraday cage.

• **Pulse Amplitude Modulation (PAM):** PAM transmits data by changing amplitudes in sequence of pulses.



(a) LoS setting (b) NLoS setting

Figure 16: Experiment settings of two scenarios: Direct Line-of-Sight and none Line-of-Sight. We change the distance between the WiFi Sender and BLE devices.



(a) NI RF testbeds (b) Faraday Cage

Figure 17: The experiment setup: NI testbeds has the ability to regenerate and process sensed signals to produce CSI. TI CC2650 BLE device varied the BLE power levels to carry information.

Two metrics are used to evaluate the performance of N-way concurrent communications system.

• **Throughput:** The number of correctly received bits per second.

• **Bit error rate (BER):** The ratio of error bits to the total number of transferred bits.

Four scenarios have been conducted to evaluate the performance of our system.

• **Faraday cage:** To test the throughput and BER of our system in an ideal scenario, we utilized a Faraday cage, which can attenuate 90 dBm of the environmental interference.

• **LoS:** The BLE and WiFi devices have the Line-of-Sight paths. To achieve this, we deployed them in the same room as shown in Figure 16(a). We also test the performance of B2W communication at a distance (the WiFi Sender and BLE devices) of 0.5, 2.5, 5, 10, 12.5 meters, respectively.

• **NLoS:** The BLE and WiFi devices do not have the Line-of-Sight paths. We deployed the BLE and WiFi devices in adjacent separate rooms as shown in Figure 16(b). We also deployed WiFi and BLE devices at 3, 5, and 7 meters, respectively, and measured the performance.

• **Multiple BLEs to WiFi communications:** To test the performance and capacity of N-way ($n > 3$) B^2W^2, we performed concurrent transmission with up to 5 pairs of BLE devices in both LoS and NLoS scenarios.

We measured the throughput and BER for DAFSK and PAM, respectively, in the above four scenarios.

5.1 Experiment Setup

Our experiment used a NI RF testbed and Texas Instruments (TI) CC2650 BLE device [3] as shown in Figure 17. Figure 17(a) shows the NI RF testbed which consists of signal digitizer (PXIe-5622), signal generator (PXIe-5652), down-converter (PXIe-5601), and up-converter (PXIe-5450). Figure 17(b) shows the CC2650 BLE devices and WiFi antennas in the Faraday cage. We use one pair of CC2650 BLE device to test our original 3-way B^2W^2 system and up to 5 pairs to test the N-way concurrent communications.

CC2650 BLE devices send packets to both the WiFi and BLE receivers concurrently. The BLE device conveys information to WiFi receiver by changing the transmission power levels of traditional BLE packets. To embed different information into WiFi packets, we modulated arbitrary bits with DAFSK and PAM schemes. The carriers for DAFSK were proportional to the transmission period of CC2650 BLE device such that the entire symbol could be sent in one seconds. Our transmitters fractionally re-sample the transmission rate and scaled to number of available power levels of CC2650 BLE devices.

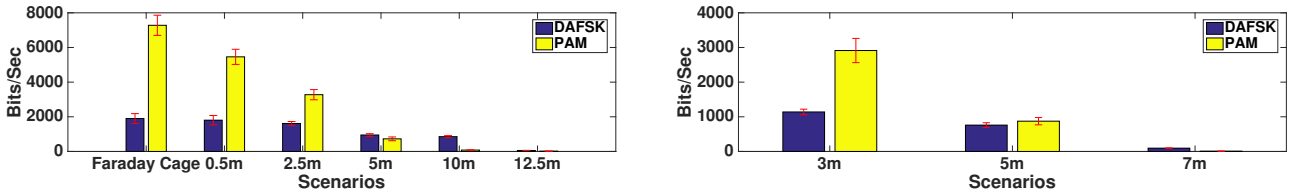
To test the performance limit of N-way concurrent communication system, we utilized a Ramsey Faraday cage to remove environmental noise. To test the performance with the various communication ranges in real world, we placed the CC2650 BLE device on a rolling chair at a 40cm height. We then changed the communication range between BLE pairs and WiFi receiver by moving the chair. Focusing on the performance of B2W communications, we fixed the distance between WiFi sender and receiver, and BLE sender and receiver. For each BER or throughput data point, we processed about 5 million WiFi data packets then calculate the average.

5.2 B2W: Throughput and BER

We evaluated the BLE to WiFi communication in a Faraday cage which provides the control of environmental interferences. Then we conduct the indoor experiments both in LoS and NLoS with different distances. We implemented our DAFSK modulation and PAM modulation. The throughput and BER in LoS and NLoS scenarios are shown as Figure 18, and Figure 19.

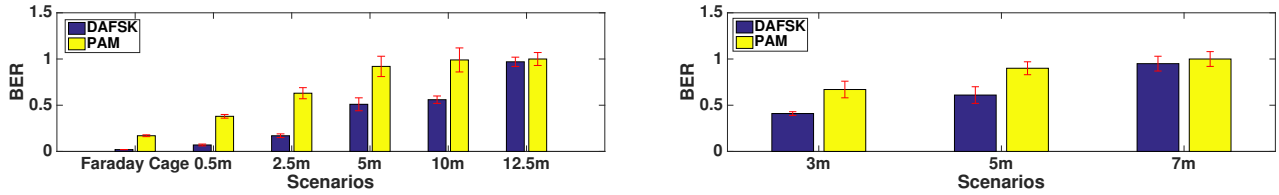
We noticed that the BER of PAM increased exponentially, but it provided 4 times more throughput. This throughput is based on the number of available power states from the BLE devices. As distance increases, the loss of sampling fidelity increases the uncertainty of the transmitted bits causing the BER to increase exponentially.

We noticed that DAFSK error rate is more linear at close distances. However, once we exceed the tolerable distance at about 10 meters, the decrease is exponential. The ability to have a higher noise tolerance is due to the fact of time based integration used in demodulation algorithms. By summing over the higher symbol duration, we achieved higher noise tolerance. Figure 18(a) shows that the PAM modulation could achieve a high throughput within 5m. However, our DAFSK scheme has a relative higher throughput at 950 bps, 855 bps, and 57 bps than PAM when the distance increased to 5m, 10m, and 12.5m, respectively. The NLoS results are shown in Figure 18(b). Due to the power attenuation and multipath effect, the throughput in LoS scenarios is up to



(a) LoS (b) NLoS

Figure 18: The throughput of BLE to WiFi devices communications in both LoS and NLoS scenarios shows that at close ranges, PAM performance is better than DAFSK. DAFSK maintains throughput better at longer distances.



(a) LoS (b) NLoS

Figure 19: The BER of BLE to WiFi devices communications shows that DAFSK resulted in fewer errors than PAM in both LoS and NLoS scenarios. While PAM has higher throughput, it is more prone to errors in noisy environments.

BLE→WiFi	B²W² PAM	B²W² DAFSK	FreeBee
Throughput	3.1kbps	1.5 kbps	17 bps

Figure 20: Compared with FreeBee under the same setting, our B^2W^2 has more than 182X or 85X throughput improvement by using PAM or DAFSK, respectively.

2.7 times than that in NLoS scenarios.

In terms of robustness, the throughput of BLE to WiFi communication drops sharply in PAM modulation scheme. However, the throughput of our DAFSK scheme remains constant when the communication range increasing from 0.5m to 10m in LoS scenario (shown in Figure 18(a)). In NLoS scenario, DAFSK also has a similar trend when the communication range increases (shown in Figure 18(b)).

The BER in NLoS scenarios is also higher than that in LoS scenarios. The BER plotted in Figure 19(a) and Figure 19(b) show that PAM suffers much more errors than DAFSK in all the LoS and NLoS scenarios.

Though our B^2W^2 system focuses on N-way concurrent communication, which is different from previous cross-technology communication, we still conducted the experiment which has the same setting with state-of-the-art work FreeBee [22]. Our result in Figure 20 shows more than 182X and 85X throughput improvement by PAM and DAFSK, respectively.

5.3 W2W: Impact of BLE Distance

We discuss the impact of BLE on W2W transmission in this section. Figure 21 shows the BER decrease sharply as distance increase between BLE and WiFi devices. As distance increase, the impact of BLE is lower. However, the BER could be further reduce by a equalization scheme (specified in Section 5.6). The experiment results show that the BER was decrease 50% after the equalization. Figure 21(a) and Figure 21(b) show that the efficiency of our equalization method works in both LoS and NLoS scenarios with different distance.

5.4 B2B: Impact of WiFi Power Levels

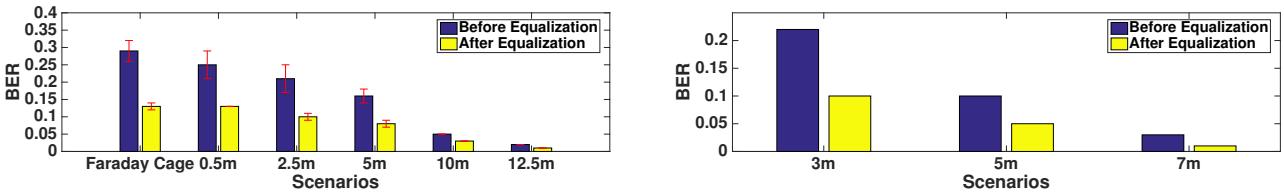
For BLE to BLE communication, the WiFi interference must be at sufficient low levels. To measure this threshold, we experimented with different received power levels of the WiFi sender, while measuring the performance of the BLE to BLE communications. Figure 23 shows the throughput and BER of BLE to BLE link with varying received power level of WiFi signals.

We use the BLE device to measure the RSSI at different distances (from WiFi sender). As distance increase, WiFi transmitting power decrease. The result shows the BLE pairs can perform transmission when RSSI lower than -19 dBm. In order to improve the throughput and minimize the BER, we demonstrate a simple WiFi interference cancellation scheme. We mixed an inverse WiFi beacon prior to BLR signal reception. Our reasoning is that standard WiFi beacons are known, and thus its phase inverse can be added during instances of interference. The BLE BER improves by 20% (Figure 23) with WiFi beacon cancellation.

5.5 Aggregated Throughput

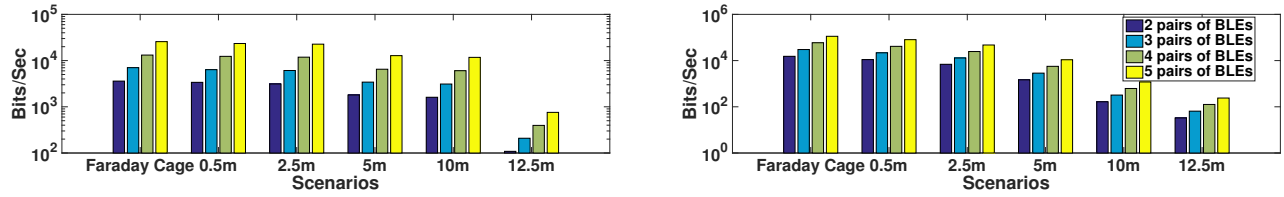
This section investigates the aggregated throughput of multiple pairs of BLE to WiFi devices communications. We demonstrate empirical transmission of 2, 3, 4 and 5 pairs of BLE devices. The throughput increases almost linearly with the increase of the BLE pairs in LoS (Figure 22) as well as NLoS (Figure 24). This is because the scalability of BLE pairs is still far away from the upper bound.

N-way concurrent communications depend on the random hopping sequences and asynchronous transmitting among pairs discussed in Section 4.2. We also simulated an analysis on the upper bound of scalability of our N-way communication system in Section 6.



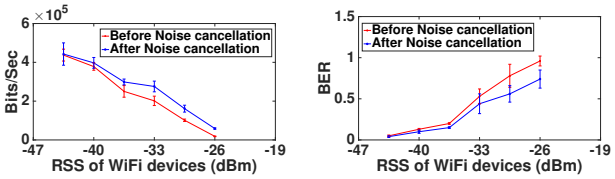
(a) LoS (b) NLoS

Figure 21: The BER of WiFi to WiFi communications with Bluetooth interference show that the shorter the distance between WiFi and BLE, the higher the BER. We demonstrate by our equalization and BLE cancellation scheme decreases BER by 50% in both LoS and NLoS scenarios.



(a) DAFSK (b) PAM

Figure 22: Throughput of up to five pairs of BLE device in LoS scenario.



(a) Throughput (b) Bit error rate

Figure 23: The throughput and BER of BLE to BLE communication show that we are able to cancel out the known WiFi beacons by using a inverted WiFi beacon phase.

5.6 Equalization

As stated in Section 5.3, the BER of W2W increases because of the presence of BLE transmission. However, because the modulation of BLE (Gaussian frequency shift keying) is consistent on amplitude. We can multiply an inverse phased BLE signal. After the multiplication, we use the correlation between neighboring CSI value to compensate the change in vector distance. Using the correction information on the affected carriers, we could decrease BER which is shown in Figure 21.

Figure 25(a) shows that some of the symbols are interfered by BLE signals. By performing the equalization scheme, Figure 25(b) demonstrates that we can correct the BLE interference when only one pair of BLE presents; while Figure 25(c) demonstrates that we also can correct the interference with five pairs of BLE. This scheme decreases BER by about 50%.

5.7 Insight Analysis

Based on the experimental data, we analyze the characteristics of the two modulation schemes (PAM and DAFSK) and discuss the impact to traditional WiFi and BLE networks.

For the two modulation schemes, we can conclude that:

• PAM provided the highest throughput if there was no interference because PAM has higher spectrum efficiency than DAFSK. For example, the constellation diagram in Figure 26(a) shows eight states in one symbol (each symbol is corresponding to one BLE packet). However, the performance of PAM reduces sharply in the real world because PAM is highly noise-interference sensitive.

• DAFSK was more robust in long distance communication because DAFSK utilizes discrete sine waves to represent bits (as introduced in Section 3.1.2) which is noise immune. For example, as shown in Figure 26(b), four sine waves with different periods are used to transmit two bits information.

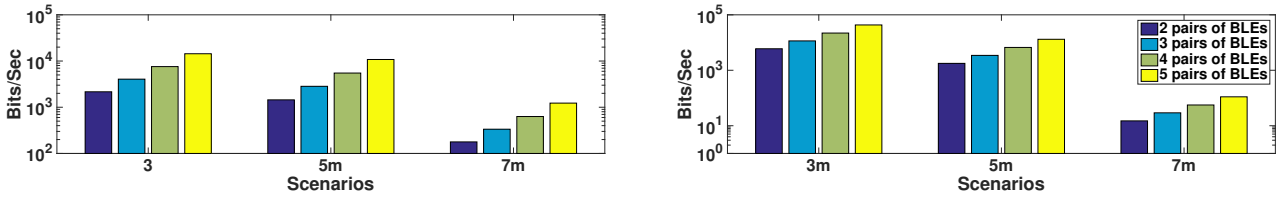
Our equalization and noise cancellation scheme leveraged known Bluetooth GFSK interference signals. The scheme reduced BER for WiFi to WiFi communication by about one half. We note that the bit errors were not just caused by Bluetooth but by multipath, Doppler Effect, but other environmental interferences as well, which are corrected by the other equalization schemes. To cancel out WiFi interference that is received by Bluetooth devices, we produce an inverted phase WiFi beacon. Because of the limited amount of beacons, the WiFi signal cancellation only decreases BER by 20%.

6. LARGE-SCALE SIMULATION EVALUATION

In order to better understand the performance of our N-way concurrent communication system B^2W^2 in large-scale scenario, we conducted extensive simulations and show the results in this section.

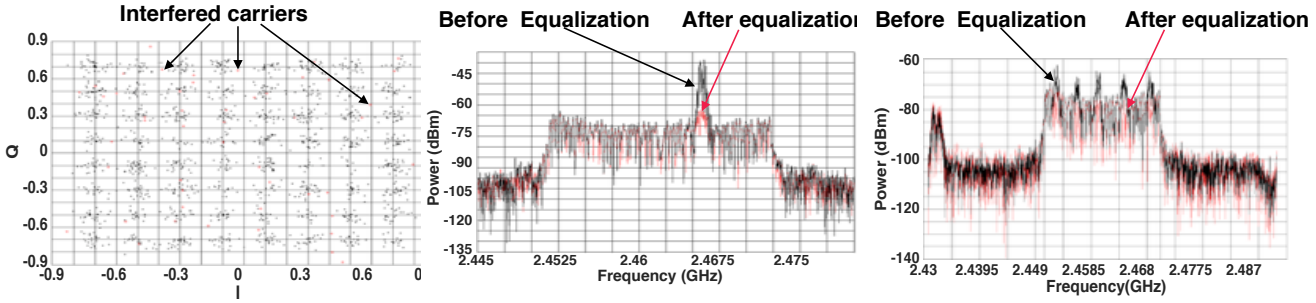
6.1 Simulation Settings

We simulate the coexistence of multiple pairs of BLE devices and WiFi Receivers. In the simulator, the BLE devices inherited the frequency hopping feature (i.e., randomly hopping among BLE Channels 0-36). The WiFi devices occupy



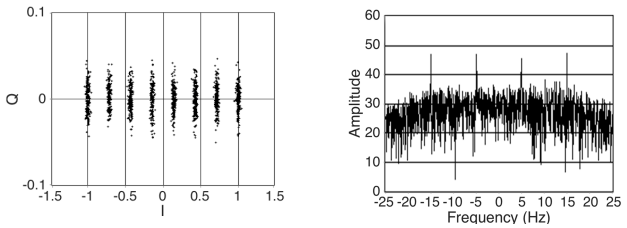
(a) DAFSK (b) PAM

Figure 24: Throughput of up to five pairs of BLE device in nLoS scenario.



(a) Constellation diagram (b) WiFi spectrum with one BLE pair (c) WiFi spectrum with multiple BLE pairs

Figure 25: These diagrams show the ability to sense interfered carriers and how the equalization scheme works. The red points in the constellation diagram are interfered carriers. The red colored spectrum demonstrate the effectiveness of the equalization and noise cancelling scheme while the black colored spectrum is the interfered spectrum.



(a) PAM (b) DAFSK

Figure 26: Two modulations produced by BLE and sensed by WiFi using CSI values. By varying the transmission power levels of BLE packets, BLE packets can carry information that can be sensed by WiFi.

either one, two, or three of the WiFi Channels 1, 6, and 11. The other variables are as follows:

- The number of BLE device increases from 10 to 50.
- The number of WiFi device is one, two, or three.
- The BLE payload lengths are: 17 Bytes (minimum payload in BLE), 32 Bytes (median payload in BLE), or 47 Bytes (maximum payload in BLE).

6.2 Simulation Results

In this section, we conclude the simulation results from three aspects: i) the aggregated throughput affected by the increasing number of BLE devices; ii) the impact on aggregated throughput of BLE payload length; and iii) the capacity of N-way concurrent communication system under different parameters.

Figure 27 shows that when the number of WiFi receivers increased, the aggregated throughput improved almost linearly. This is reasonable since the aggregated WiFi bandwidth is almost trebled as shown in Figure 14.

The payload length does not impact the aggregated throughput significantly when the number of BLE pairs is less than 8. This is because that the frequency hopping scheme yields a low collision possibility when the number of BLE pair is small. The growing trend of throughput decreases with the pair number range from 10 to 22 . After the number reaches 22, the collision occurs frequently which causes the throughput begin to drop. Because the collision possibility of small payload length is less than large payload length, and in our B^2W^2 system, each BLE packet only carries one symbol which will not affect by the payload length, so the aggregated throughput increases. However, with the reduced payload, one tradeoff is the synchronization on the WiFi side should be more accurate since the small packet is hard to capture.

Based on the simulation results, we draw the following conclusion: the maximum throughput of N-way concurrent communication occurs when there are 22 pairs of BLE device with a payload of 17 Byte and three WiFi receivers.

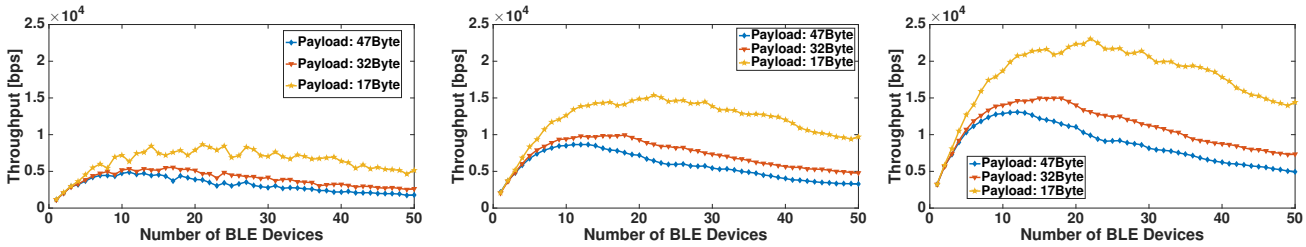
7. RELATED WORK

The related work can be divided into three categories: single-technology, hybrid-technology, and cross-technology communications.

7.1 Single-Technology Communication

Within this category, the most related technologies are Bluetooth, Bluetooth Low Energy (BLE), and WiFi.

Bluetooth uses frequency hopping to avoid the collision. It



(a) 1 WiFi receiver (b) 2 WiFi receivers (c) 3 WiFi receivers

Figure 27: The throughput with different number of BLE senders and WiFi receivers

has been widely deployed in smartphones, laptops and sensor nodes [31, 23, 49, 37] to support some IoT applications such as localization [14], health assisting [47], and social sensing [11]. **BLE** is inteneded to reduce power consumption while maintaining a similar communication range as the classical Bluetooth, which makes BLE particularly suitable for energy-sensitive IoT applications, such as human behavior sampling [47], wearable devices [4, 38], localization [24], and low-energy devices [9]. However, as researchers pointed out that BLE-based IoT devices have a gateway problem [48], which is one of the motivations for us to conduct this project that can minimize the impact of the gateway problem on BLE devices. **WiFi** has been widely deployed and allows electronic devices (including IoT devices [32]) connect to a wireless local area network. It supports the applications such as tracking [30, 33], remote monitoring and surveillance [19, 8]. The channel state information (CSI) of WiFi has been used for evaluating and analyzing communication channels [12, 39, 6] and also utilized for many applications such as activity recognition [42].

Build on top of the above communication technologies, researchers have investigated various routing metrics [43] and protocols for load balancing [44], enhancing network reliability [5], strengthening network security [45], achieving faster routing with lower cost [35, 7], reducing energy consumption [16, 41] and other resources in the network [29, 40]. Different from the above approaches, our design enables *concurrent* cross-technology communication.

7.2 Hybrid-Technology Communication

In this category, at least two communication technologies are combined together for better performance. For example, researchers investigated how to utilize heterogeneous radios for higher throughput and lower end-to-end delay in [17, 18]. Howies [51] uses a ZigBee to wakeup the WiFi radio to save energy in mobile devices. Backscatter techniques [28, 34, 27] have recently revolutionized the wireless communication. The device harvests the energy from RF signal and embeds the message into the reflected the RF signal. Wi-Fi Backscatter [20] can harvest energy from WiFi and reuse existing WiFi infrastructure to provide internete connectivity to RF-powered devices. Passive Wi-Fi [21] can significantly improve the data rate and distance by introducing a plugged-in device.

Different from the above approaches in this category, the main design goal of our approach is to enable the *concurrent* communications of multiple IoT devices without introducing extra hardware and traffic.

7.3 Cross-Technology Communication

Instead of mitigating the interference [25], researchers proposed to leverage the coexistence feature of multiple communication technologies under the same frequency for cross-technology communication [10, 50, 22]. Received Signal Strength (RSS) has been used for cross-technology communication. RSS is an important indicator of communication ambient and has been widely applied to fields such as bearing estimation [13], monitoring network devices [26], and improving MAC layer design [36]. In Esense [10], GSense [50] and FreeBee [22], researchers use RSS to measure the WiFi signal so that it enables the communication between WiFi and ZigBee devices. Using FreeBee, the BLE to WiFi communication's throughput is 17 bps (based on Figure 17 in [22]). To solve the IoT devices' gateway problem [48], researchers developed crosstalk based primitive to enable communication between WiFi devices and IEEE 802.15.4 devices and achieved a data rate of 2 bytes per second [46].

Different from the above approaches, our design enables multiple BLE devices *concurrently* communicate different pieces of information with multiple WiFi devices and achieve orders of magnitude higher throughput for individual node pairs than existing approaches.

8. CONCLUSION

In this paper, we introduce an N-way *concurrent* communication framework (B^2W^2), which can concurrently conduct three different types of communications among multiple IoT devices equipped with WiFi or BLE. To put it in a nutshell, B^2W^2 enables the high throughput and long distance *concurrent* N-way cross-technology communication by leveraging channel state information (CSI). Our design is possible to provide another gateway for the BLE-based IoT devices. We investigated a significant amount of effort to evaluate our design under one ideal setting and three different real-world settings. Our empirical results demonstrate that we can achieve more than 85X times higher throughput compared with the most recently reported cross-technology protocol (i.e., FreeBee [22]).

N-way *concurrent* communication opens a new door to more efficiently utilize the spectrum under pressure of the exponentially increasing number of IoT devices and the huge amount of data generated by these devices.

9. ACKNOWLEDGMENTS

This project is supported by NSF grants CNS-1503590 and CNS-1539047. We also thank anonymous reviewers and Dr. Anthony Rowe for their valuable comments.

10. REFERENCES

[1] http://cloudtimes.org/2013/12/20/gartner-the-internet-of-things-will-grow-30-times-to-26-billion-by-2020/.

[2] https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=237781/.

[3] http://www.ti.com/lit/ds/symlink/cc2650.pdf.

[4] M. Alessandro, N. Sarfraz, and M. Cecilia. Experience in deploying wearable devices for office analytics. *CSCW, 2016*.

[5] A. B. M. Alim Al Islam and V. Raghunathan. itcp: an intelligent tcp with neural network based end-to-end congestion control for ad-hoc multi-hop wireless mesh networks. *Wireless Networks, 2014*.

[6] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: A survey. *TOSN, 2012*.

[7] R. Banirazi, E. Jonckheere, and B. Krishnamachari. Dirichlet's principle on multiclass multihop wireless networks: Minimum cost routing subject to stability. In *MSWiM, 2014*.

[8] M. Buevich, D. Schnitzer, T. Escalada, A. Jacquiau-Chamski, and A. Rowe. Fine-grained remote monitoring, control and pre-paid electrical service in rural microgrids. In *IPSN, 2014*.

[9] B. Campbell, J. Adkins, and P. Dutta. Cinamin: A perpetual and nearly invisible ble beacon. In *EWSN, 2016*.

[10] K. Chebrolu and A. Dhekne. Esense: Communication through energy sensing. In *MobiCom, 2009*.

[11] Z. Chen, Y. Chen, X. Gao, S. Wang, L. Hu, C. C. Yan, N. D. Lane, and C. Miao. Unobtrusive sensing incremental social contexts using fuzzy class incremental learning. In *ICDM, 2015*.

[12] R. Crepaldi, J. Lee, R. Etkin, S.-J. Lee, and R. Kravets. Csi-sf: Estimating wireless channel state using csi sampling and fusion. In *INFOCOM, 2012*.

[13] K. Dantu, P. Goyal, and G. S. Sukhatme. Relative bearing estimation from commodity radios. In *ICRA, 2009*.

[14] T. Guan, W. Dong, D. Koutsonikolas, G. Challen, and C. Qiao. Robust, cost-effective and scalable localization in large indoor areas. *GLOBECOM, 15*.

[15] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with channel state information. *SIGCOMM Comput. Commun. Rev.*, 41(1):53–53, Jan. 2011.

[16] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance system using wireless sensor networks. In *MobiSys, 2004*.

[17] A. B. M. A. A. Islam and V. Raghunathan. Siac: simultaneous activation of heterogeneous radios in high data rate multi-hop wireless networks. *Wireless Networks, 2015*.

[18] A. B. M. A. A. Islam and V. Raghunathan. Symco: Symbiotic coexistence of single-hop and multi-hop transmissions in next-generation wireless mesh networks. *Wireless Networks, 2015*.

[19] A. Kandhalu, A. Rowe, R. Rajkumar, C. Huang, and C. C. Yeh. Real-time video surveillance over ieee 802.11 mesh networks. In *RTAS, 2009*.

[20] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall. Wi-fi backscatter: Internet connectivity for rf-powered devices. In *SIGCOMM, 2014*.

[21] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith. Passive wi-fi: Bringing low power to wi-fi transmissions. In *NSDI, 2016*.

[22] S. M. Kim and T. He. Freebee: Cross-technology communication via free side-channel. In *MobiCom 2015*.

[23] R. Kling, R. Adler, J. Huang, V. Hummel, and L. Nachman. Intel mote-based sensor networks. *Struct. Control Health Monit*, 2005.

[24] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe. Alps: A bluetooth and ultrasound platform for mapping and localization. In *SenSys, 2015*.

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

[26] R. Lim, M. Zimmerling, and L. Thiele. Passive, privacy-preserving real-time counting of unmodified smartphones via zigbee interference. In *DCOSS, 2015*.

[27] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient backscatter: Wireless communication out of thin air. In *SIGCOMM, 2013*.

[28] V. Liu, V. Talla, and S. Gollakota. Enabling instantaneous feedback with full-duplex backscatter. In *MobiCom, 2014*.

[29] C. Mascolo and M. Musolesi. Scar: Context-aware adaptive routing in delay tolerant mobile sensor networks. In *IWCMC, 2006*.

[30] A. B. M. Musa and J. Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *SenSys, 2012*.

[31] L. Nachman, R. Kling, R. Adler, J. Huang, and V. Hummel. The intel mote platform: A bluetooth-based sensor network for industrial monitoring. In *IPSN, 2005*.

[32] B. Ostermaier, M. Kovatsch, and S. Santini. Connecting things to the web using programmable low-power wifi modules. In *WoT 2011*, San Francisco, CA, USA, June 2011.

[33] S. Papaioannou, H. Wen, Z. Xiao, A. Markham, and N. Trigoni. Accurate positioning via cross-modality training. In *SenSys, 2015*.

[34] A. N. Parks, A. Liu, S. Gollakota, and J. R. Smith. Turbocharging ambient backscatter communication. In *SIGCOMM, 2014*.

[35] A. Reinhardt, O. Morar, S. Santini, S. Zöller, and R. Steinmetz. Cbfr: Bloom filter routing with gradual forgetting for tree-structured wireless sensor networks with mobile nodes. In *WoWMoM, 2012*.

[36] M. Ringwald and K. Römer. Burstmac-low idle overhead and high throughput in one mac protocol. *EWSN 2008*.

[37] M. Ringwald and K. Romer. Practical time synchronization for bluetooth scatternets. In *BROADNETS 2007*.

[38] R. C. Shah, L. Nachman, and C.-y. Wan. On the performance of bluetooth and ieee 802.15.4 radios in a

body area network. In *BodyNets, 2008*.

[39] J. Shi, L. Meng, A. Striegel, C. Qiao, D. Koutsonikolas, and G. Challen. A walk on the client side: Monitoring enterprise wifi networks using smartphone channel scans. In *INFOCOM, 2016*.

[40] L. Su, B. Ding, Y. Yang, T. F. Abdelzaher, G. Cao, and J. C. Hou. ocast: Optimal multicast routing protocol for wireless sensor networks. In *ICNP, 2009*.

[41] M. Y. S. Uddin, H. Ahmadi, T. Abdelzaher, and R. Kravets. Intercontact routing for energy constrained disaster response networks. *IEEE Transactions on Mobile Computing, 2013*.

[42] B. Wei, W. Hu, M. Yang, and C. T. Chou. Radio-based device-free activity recognition with radio frequency interference. In *IPSN, 2015*.

[43] Y. Yang, J. Wang, and R. Kravets. Designing routing metrics for mesh networks. In *WiMesh, 2005*.

[44] Y. Yang, J. Wang, and R. Kravets. Load-balanced routing for mesh networks. *SIGMOBILE, 2006*.

[45] S. Yi, P. Naldurg, and R. Kravets. Security-aware ad hoc routing for wireless networks. In *MobiHoc, 2001*.

[46] S. Yin, Q. Li, and O. Gnawali. Interconnecting wifi devices with ieee 802.15.4 devices without using a gateway. In *DCOSS, 2015*.

[47] C.-w. You, K.-C. Wang, M.-C. Huang, Y.-C. Chen, C.-L. Lin, P.-S. Ho, H.-C. Wang, P. Huang, and H.-H. Chu. Soberdiary: A phone-based support system for assisting recovery from alcohol dependence. In *CHI, 2015*.

[48] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta. The internet of things has a gateway problem. In *HotMobile, 2015*.

[49] X. Zhang, L. Nachman, G. F. Riley, and R. Kling. Metric-based scatternet formation and recovery optimization for intel mote. In *Mobile and Ubiquitous Systems, 2006*.

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

[51] Y. Zhang and Q. Li. Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices. In *INFOCOM, 2013*.