



# FreeBee: Cross-technology Communication via Free Side-channel

Song Min Kim and Tian He

Department of Computer Science and Engineering, University of Minnesota, USA  
{ksong, tianhe}@cs.umn.edu

## ABSTRACT

This paper presents FreeBee, which enables direct unicast as well as cross-technology/channel broadcast among three popular wireless technologies: WiFi, ZigBee, and Bluetooth. Our design aims to shed the light on the opportunities that cross-technology communication has to offer including, but not limited to, cross-technology interference mitigation and context-aware smart operation. The key concept of FreeBee is to modulate symbol messages by shifting the timing of periodic beacon frames already mandatory for wireless standards without incurring extra traffic. Such a generic cross-technology design consumes zero additional bandwidth, allowing continuous broadcast to safely reach mobile and/or duty-cycled devices. A new *interval multiplexing* technique is proposed to enable concurrent broadcasts from multiple senders or boost the transmission rate of a single sender. Theoretical and experimental exploration reveals that FreeBee offers a reliable symbol delivery under a second and supports mobility of 30mph and low duty-cycle operations of under 5%.

## Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless communications

## Keywords

Wireless; Protocol; Cross-technology

## 1. INTRODUCTION

The unprecedented proliferation of wireless devices in the last decade has been led by multiple wireless technologies, each offering convenience in different aspects of our daily lives. However, wireless technologies are victims of their own success: spectrum sharing among incompatible wireless technologies has led to a severe *wireless coexistence problem* [18, 28, 33, 38, 39, 42].

This paper begins with the recognition that this coexistence is indeed double side: although it may cause inefficiency and unfairness in channel/spectrum utilization, it

also provides new opportunities because the standards for individual technologies are specialized and hence possess strengths in different areas that are, often the weaknesses of the others. For example, while WiFi has access to a virtually unlimited amount of information via the Internet, it consumes a considerable amount of power, causing battery problems in mobile devices [10, 16]. Conversely, the ZigBee network often operates as a stand-alone and has limited information, but is extremely energy efficient. Thus, both networks can be enhanced via mutual supplementation, demonstrating the positive side of coexistence.

In this paper, we propose FreeBee, a cross-technology communication framework that is generic and transparent (i.e., no extra traffic). Our design aims to mitigate the detrimental effect of coexistence while exploring the opportunities behind it. As such, FreeBee sheds the light on the opportunities that cross-technology communication has to offer including, but not limited to, cross-technology interference mitigation and context-aware smart operation. Specifically we achieve this via embedding symbol into beacons by shifting their transmission timing. Although the concept of modulating via signal timings is known as PPM (Pulse Position Modulation), legacy PPM supports only communication between homogeneous devices and requires precise pulse timing, which can be hardly satisfied in wireless coexistence environments with mainly contention-based MACs.

Existing cross-technology communication works [14, 40] are technology-specific and require dedicated packets for communication, burdening already-crowded channels with further overhead. In contrast, FreeBee utilizes mandatory beacons widely adopted among wireless technologies [5, 35, 36], achieving a generic and free-side-channel design. In summary, our original contribution is three-fold:

- We propose FreeBee, a novel cross-technology communication framework that allows direct communication between heterogeneous senders and receivers. In addition, FreeBee allows heterogeneous devices to receive broadcast simultaneously from a sender with overlapping frequencies (e.g., Bluetooth to WiFi and ZigBee) and support a sender with a wider bandwidth (e.g., WiFi) to reach multiple narrower-band receivers (e.g., WiFi to multi-channel ZigBee).
- FreeBee requires no hardware modification and does not introduce dedicated traffic. Its existence is transparent to legacy wireless systems. Our new *interval multiplexing scheme* supports concurrent transmission and reception of multiple signals.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

MobiCom'15, September 7–11, 2015, Paris, France.

© 2015 ACM. ISBN 978-1-4503-3619-2/15/09 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2789168.2790098>.

- We present three prototype implementations: WiFi, ZigBee, and Bluetooth. Results suggest that FreeBee offers reliable symbol delivery within less than a second and supports mobility up to 30mph and duty cycle operation of under 5%. We also demonstrate a practical use of FreeBee: Inspecting real WiFi deployment patterns in a shopping mall area, FreeBee was found to save 78.9% of the energy otherwise wasted by the WiFi interface.

The paper is organized as follows. Section 2 presents motivations. The FreeBee design and features are introduced in Sections 3 and 4. Sections 5 and 6 evaluate performance analytically and empirically, followed by a FreeBee application in Section 7. We summarize related work in Section 8 and conclude in Section 9.

## 2. MOTIVATION

This section demonstrates a wide range of benefits the FreeBee technology has to offer.

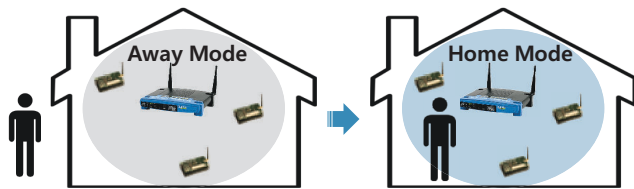


Figure 1: Context-aware home automation

- **Benefit to ZigBee – Smart homes:** FreeBee enables direct sharing of information without assistance from costly (>150 USD) dual-radio gateways that are mostly unavailable in real-life settings. One such type of information is user presence, which is accessible by WiFi AP by observing the nodes associated to it. Sharing this information enables other networks to provide context-aware service. Figure 1 demonstrates the example of a smart home with ZigBee-assisted appliances. Home WiFi AP first determines whether the resident is away or home (i.e., his/her smart phone is associated or not). Using FreeBee, this information is broadcast from the AP to all the ZigBee nodes inside the home to drive them to the appropriate operation mode. For instance, once the resident leaves they turn to “away mode”, such as lowering home temperature to an energy-economic value.

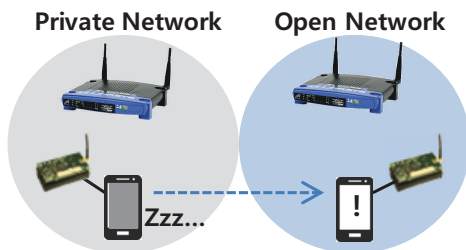


Figure 2: Wake on selective WLAN

- **Benefit to WiFi – Mobile devices:** Operating a WiFi Network Interface Card (NIC) continuously depletes precious energy in portable devices [10, 16]. To tackle this, ZifFi [43] suggests attaching a low-power ZigBee radio to the device to wake up the WiFi NIC whenever it detects the existence of *any* WiFi AP. While this approach can significantly

reduce standby energy, we believe that further savings can be achieved.

Nowadays most of metropolitan areas are overloaded with WiFi APs, many of which are private. Thus, it is still a waste of energy to blindly wake up a WiFi NIC when an arbitrary AP shows up, without knowing if it is accessible or not. As shown in Figure 2, we avoid this issue by embedding 1-bit accessibility information (i.e., open/private) into WiFi beacons and allowing the attached low-power radio (Bluetooth or ZigBee) to capture this information through FreeBee. Accordingly, the WiFi NIC can wake up only when it finds an open AP. We refer to this function as *wake on selective WLAN*. We note that this approach can be extended to limiting WiFi wakeup only to discovering an AP that fits the user’s interest; for instance, when there is an AP that matches a user-defined SSID.

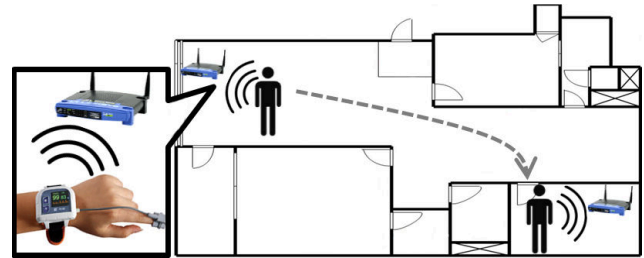


Figure 3: Real-time patient monitoring

- **Benefit to Bluetooth – Health care:** Taking advantage of its low-power operation, Bluetooth technology is widely used in portable medical devices, including glucose and heart monitors [13]. Although FreeBee is not designed to transfer a large volume of medical data, it enables health alerts by embedding urgent information into Bluetooth beacons. The ubiquitous WiFi coverage in most indoor environments today provides a continuous alert service even if patients are away from their Bluetooth-enabled medical station, as shown in Figure 3. This figure also shows that FreeBee can offer the location of the patient via geolocation provided by WiFi AP, allowing accurate and timely medical actions in case of an emergency.



Figure 4: Cross-technology coordination

- **Benefit to all – Channel efficiency:** All WiFi, ZigBee, and Bluetooth networks can benefit from FreeBee via cross-technology channel coordination. As a result of evolving separately without considering each other, channel access schemes in heterogeneous wireless technologies are incompatible, leading to a severe CTI [20, 25, 40, 41]. This cross-technology channel access problem can be addressed by allowing explicit communication among different technologies. As demonstrated in Figure 4, FreeBee essentially allows TDMA or FDMA among heterogeneous wireless platforms, alleviating CTI. For instance, FreeBee realizes mechanisms similar to NAV (Network allocation vector) or RTS/CTS in WiFi for spectrum allocation that is global across technologies.

### 3. FRAMEWORK DESIGN

This section presents an overview of FreeBee, followed by design specifics such as modulation and demodulation techniques. Without loss of generality, we use communication between WiFi (sender) and ZigBee (receiver) to illustrate the generic design of FreeBee.

#### 3.1 Design Overview

We propose a cross-technology communication framework in which symbols are embedded into the timing of beacon frames. Specifically, we slightly shift the transmission time (advance or delay) of beacon frames, a configurable setting in most WiFi APs deployed today, simply via HTTP protocol. The shift is made in the units of 1.024ms, compliant to the 802.11 standard unit used in beacon scheduling, known as TBTT (Target Beacon Transmission Time) [35]. To ensure a free side-channel operation, FreeBee shifts timing in such a way that the average interval remains the same as the original setting. Thus, the proposed communication framework is not only transparent, but also does not consume additional bandwidth or energy. This unique aspect enables important information to be broadcast continuously, safely reaching mobile and/or duty-cycled ZigBee receivers whose presence or active periods are a priori unknown to the sender.

However, WiFi beacons cannot be directly captured and recognized by ZigBee nodes, due to incompatible PHY layers. Instead, as WiFi coexists with ZigBee on the 2.4GHz ISM band, a ZigBee receiver statically detects the position of the WiFi beacons in the wireless channel, using its RSSI sensing capability. We note that, as a foundational function for MAC techniques including CSMA, RSSI sampling is common among varieties of wireless standards (e.g., Bluetooth).

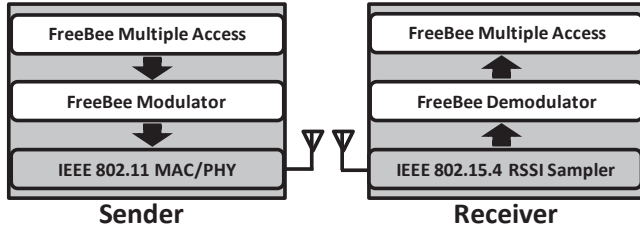


Figure 5: Architecture and scope in white boxes

Figure 5 depicts the overall architecture and the scope of this paper. Our design spans modulation/demodulation techniques, as well as a multiple access scheme for concurrent communications. We note that the WiFi-ZigBee FreeBee design is based on 802.11 and 802.15.4 standards, and our design is compatible with all 802.11 variants (i.e., a/b/g/n) and 802.15.4-compliant nodes (e.g., TelosB and MICAz) and requires no hardware modification. In fact, FreeBee can be adopted to enable communication between any heterogeneous wireless platforms as long as the channel is shared. In the following, we first propose a basic version of FreeBee with the simplest form, followed by elaborated designs that enhance the basic version.

#### 3.2 Basic FreeBee

In this section, we describe the basic version of modulation/demodulation we refer to as *FreeBee*. This design assumes that the unmodulated position of the beacon is found

during the initial network setup, which we refer to as the *reference position*. This position can be simply obtained by running FreeBee when the AP is sending beacons in their original timing. Modulated beacon positions found later are compared to the reference position for symbol interpretation (i.e., demodulation).

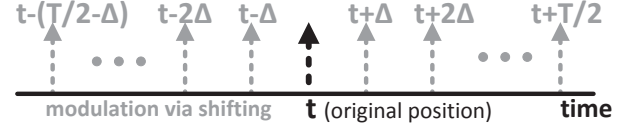


Figure 6: The symbol is embedded by shifting the beacon from its reference position,  $t$ , where the degree of shift ranges between  $(-T/2, T/2]$ .

##### 3.2.1 Modulation

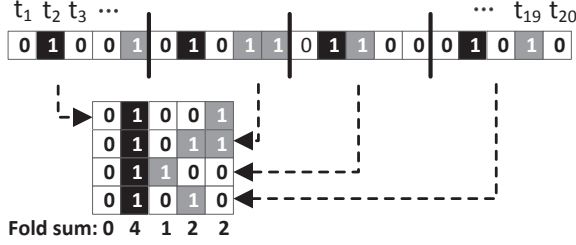
For timely advertisement, the 802.11 standard requires APs to periodically broadcast beacons. FreeBee establishes a free side-channel by embedding symbols within the *transmission timing* of these mandatory packets. Referring to Figure 6, let's consider a beacon whose reference position is at  $t$ , where the interval is  $T$ . Applying FreeBee, we *shift* the beacon from its reference position in the range of  $(-T/2, T/2]$  to indicate the symbol to be delivered. The amount of information that can be embedded is determined by  $T$  and the granularity of shift, indicated by  $\Delta$  in Figure 6. We set  $\Delta$  as 1.024ms following the beacon scheduling granularity as defined by the 802.11 standard (We note that the information amount can easily be increased by adopting smaller  $\Delta$ ). Under this setting the typical  $T$  of 102.4ms, adopted in the majority of legacy WiFi APs, corresponds to  $100\Delta$ , indicating that the beacon can be positioned at 100 different time instances. Thus, beacon shift can express  $\lceil \log_2 100 \rceil = 6$  bits.

Due to incompatible PHY layers, the ZigBee receiver is unable to decode the beacons and thus cannot detect the presence of beacons directly. Therefore we statistically locate beacons by their periodic repetition. For instance, to deliver a FreeBee symbol corresponding to  $t - \Delta$  in Figure 6, multiple consecutive beacons are shifted for the same amount (i.e., beacons are transmitted at  $t+T-\Delta$ ,  $t+2T-\Delta$ , and so on). The required number of beacon repetitions per symbol is decided by the channel noise, which is analyzed in detail in later sections. Lastly, we note that the beacon interval still remains at  $T$  in the process of FreeBee transmission, indicating free side-channel operation.

##### 3.2.2 Demodulation

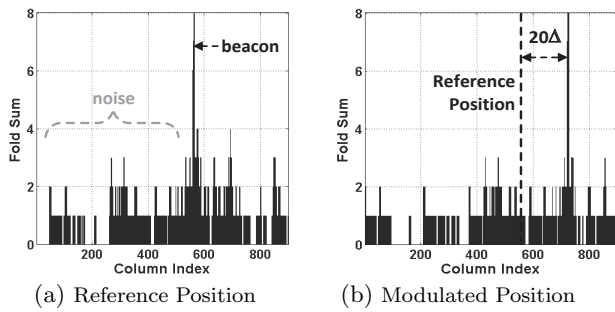
Here we describe how FreeBee captures and interprets position-modulated beacons for successful demodulation, especially under channel noise (i.e., other ongoing traffic within the same band). FreeBee demodulation starts from sampling the energy in the channel. This is done by consecutively recording the values obtained from the RSSI register on an 802.15.4-compliant RF chip on the ZigBee node. Upon recording a stream of RSSIs, the captured values are quantized to binaries—0 if below threshold and 1 if above—to indicate clear and busy channels. The threshold is set to be -75dBm following the CCA (Clear Channel Assessment) threshold for the 802.15.4 standard [36]. We note that WiFi also runs CCA, where the threshold is -82dBm [35]. For simplicity, we hereafter will refer to the binary value sim-

ply as RSSI. Furthermore, as a RSSI sample in ZigBee is a measurement spanning for  $128\mu s$ , the sampling rate is set to be  $1/128\mu s = 7.8KHz$  to avoid time gaps between samples while keeping the rate at its minimum.



**Figure 7: Folding example:** series of RSSI samples expressed as boxes. Both black and gray indicate a busy channel where the former is a periodic beacon signal with  $\lambda = 5$ , while the latter is random noise induced by traffic or interference. White represents an idle channel. By *folding* the series into a matrix with  $P = 5$  ( $= \lambda$ ), the black boxes align column-wise.

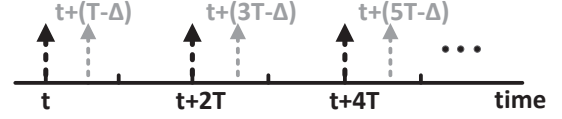
We then apply *folding* to the obtained RSSI vector, a signal processing technique that allows detecting periodic signal under noise. We note that this technique was originally introduced in [34] and was recently featured by ZiFi [43] to detect the presence of WiFi AP. Given a sampled RSSI vector, folding by  $P$  simply cuts the vector into sub-vectors of equal lengths of  $P$  and stacks them to yield a matrix. An example of folding is shown in Figure 7, where a sampled RSSI vector of length 20 is considered. Let the interval of the beacons captured in the vector be  $T$  seconds, and the number of samples in  $T$  as  $\lambda$ . In the example  $\lambda = 5$ , and upon folding by  $P = \lambda$ , RSSIs of beacons (in black) align in a column. the column-wise sum is referred to as the *fold sum*, where the column with the highest fold sum indicates the position of the beacon. Note that the fold sums are likely to be smaller in other columns, as they are induced by either random (thus aperiodic) traffic or beacons with different intervals.



**Figure 8: Example of FreeBee demodulation in practice, when  $T = 113\Delta = 115.7ms$  and  $\lambda = 904$**

Figure 8 presents an example of demodulating FreeBee symbol ( $20\Delta$ ) in practice. To sum up, FreeBee demodulation is process of finding the column corresponding to beacon position, which can easily be achieved by folding and simply picking the column with the maximum fold sum. This same process is used to learn the reference position of the beacon during network initialization and to find the modulated position. The difference of the two positions indicates the

symbol within. Other harmonic analysis techniques, such as FFT (Fast Fourier Transform) and autocorrelation, do not yield the position (i.e., phase) of the beacon, and hence are not suitable for our purpose.



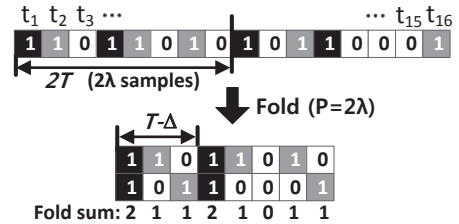
**Figure 9: A-FreeBee:** the positions of *every other* beacons are shifted (in gray), whereas the degree of shift is  $\Delta$  in the example.

### 3.3 Enhanced Feature #1: Asynchrony

The basic FreeBee design embeds the symbol as a beacon time shift from the reference position, a concept that requires learning the position beforehand. We relax this condition to introduce A-FreeBee (Asynchronous FreeBee), freeing our design from any prior knowledge to offer instantaneous communication.

Figure 9 shows Modulation of A-FreeBee. Applying A-FreeBee to beacons with an interval of  $T$ , beacons are scheduled to construct two streams of beacons (black and gray) with the same interval of  $2T$ , where one stream (i.e., gray beacons) is a shifted version of the other (i.e., black beacons). This is achieved by shifting *every other* beacon by the amount that corresponds to the symbol to be delivered. The figure demonstrates a case where the symbol corresponds to  $\Delta$ , indicating one unit of shift. We note that A-FreeBee is also a free-side-channel, as the average interval between consecutive beacons is still  $T$ .

Under A-FreeBee design, the reference position is no longer required; it simply looks for two beacon streams with the same period by folding with  $P = 2\lambda$ . The embedded symbol is interpreted directly from the phase difference, i.e., *Two* columns with the first and second highest fold sums are found, where the distance between them indicates A-FreeBee symbol. A demodulation example of A-FreeBee is



**Figure 10: Folding example for A-FreeBee:** two high fold sum columns are detected via folding.

shown in Figure 10 for a RSSI vector of length 16. Two beacon streams are depicted in black boxes, and the gray boxes represent noise. Two columns with high fold sums are found by folding with  $P = 2\lambda = 8$ , where the distance between the two columns is  $T - \Delta$ . Noting that the distance would simply be  $T$  before modulation, the amount of shift (i.e., the symbol) is therefore  $\Delta$ . Figure 11 below demonstrates A-FreeBee demodulation in practice, where the conveyed symbol is  $20\Delta$ .

The asynchronous version features several advantages over the basic FreeBee, including that it (i) requires no synchro-



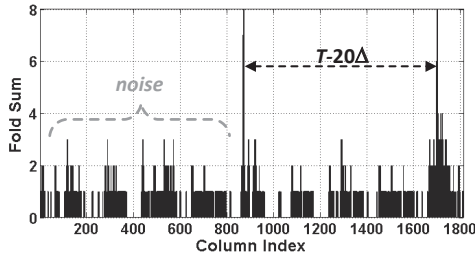


Figure 11: A-FreeBee demodulation ( $T = 113\Delta$ )

nization, (ii) is robust to clock drifts, and (iii) supports instantaneous communication without prior knowledge. We note that all these improvements come with a trade-off in data rate; A-FreeBee, compared to the basic FreeBee, requires collecting a higher number of beacons to form *two* high fold sums instead of one. The relationship between the performances of FreeBee and A-FreeBee are analyzed both theoretically and empirically in later parts of the paper.

### 3.4 Enhanced Feature #2: Concurrency

In the face of multiple (A-)FreeBee senders, selecting the same or arbitrary intervals may lead the signals to tangle and cause errors in demodulation. In this section, we address this issue to allow *simultaneous transmissions* of an arbitrary number of (A-)FreeBee symbols such that each of them can be safely demodulated at the receiver.

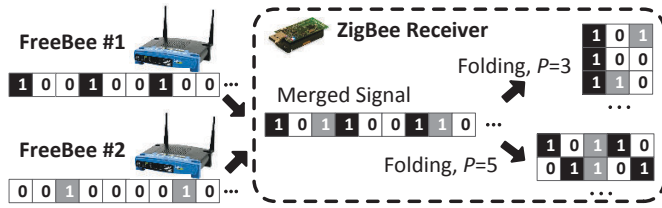


Figure 12: Demodulating interval-multiplexed FreeBee symbols.

#### 3.4.1 Interval Multiplexing

Recall that, according to the 802.11 standard, beacon intervals are defined in the unit of  $\Delta (= 1.024ms)$ . Let beacon intervals of  $n$  neighboring APs be  $x_1\Delta, x_2\Delta, \dots, x_n\Delta$ . Then, FreeBee allows simultaneous communication of  $n$  APs if  $x_1, x_2, \dots, x_n$  are pair-wise co-primes. We refer to this as *interval multiplexing*.

Figure 12 demonstrates a scenario of interval multiplexing/demultiplexing, where two FreeBee senders with intervals  $T_1 = 3\Delta$  (in black) and  $T_2 = 5\Delta$  (in gray) introduce a vector of merged RSSI signals at a receiver, and this receiver utilizes interval de-multiplexing to demodulate. Specifically, by folding with  $P = 3$  and  $5$  and looking for the column with the highest fold sum, the receiver can detect the position of the beacons as if the other signal does not exist. This is because  $3$  and  $5$  are co-primes, and no longer holds when they are not; for example, consider a sampled RSSI vector including beacons with intervals  $T_1 = 2\Delta$  and  $T_2 = 4\Delta$ . When folded by  $P = 4$ , both beacons will form high fold sums, causing demodulation error. We note that while the figure shows only two senders for clarity, this idea can be extended to  $n$  senders as long as the intervals are pairwise co-prime. The rationale behind this scheme is given in the following proposition:

**PROPOSITION 3.1.** *For FreeBee signals with co-prime intervals, folding for one signal restricts the fold sum of the others to a maximum of 1.*

**PROOF.** Let  $T_1 = x_1\Delta$  and  $T_2 = x_2\Delta$  be two beacon intervals where  $x_1$  and  $x_2$  are co-primes. When a sampled RSSI vector containing beacons of interval  $T_2$  is fold by  $P = x_1$ , beacon repeats in a column every LCM (least common multiple) of  $x_1$  and  $x_2$ , which is  $x_1 \times x_2$ . Since the total length of the sampled RSSI vector is much smaller than  $x_1 \times x_2$ , beacons with the interval of  $x_1\Delta$  cannot be folded into the same column when folded by  $x_2$  (and vice versa), yielding the maximum fold sum of 1.  $\square$

The proposition states that the cross-interference between FreeBee signals is effectively suppressed when intervals are co-prime, essentially granting orthogonality between signals for concurrent communication. We note that this holds for both basic FreeBee and A-FreeBee.

In practice, to avoid the overhead in computing for co-prime numbers, we allow APs to select among a set of pre-stored *prime* numbers instead. Moreover, the same interval should not be chosen among neighboring APs, which can be easily achieved in *listen before talk* fashion; they listen to each other to acquire the beacon interval information (i.e.,  $x$  in  $T = x\Delta$ ) that, according to the 802.11 standard, is recorded within beacons. After storing a set of intervals used by the neighboring APs, an unoccupied prime number is chosen as its interval. Conversely, coordination via wired connection (i.e., WLAN or the Internet) may be preferred, which avoids the hidden terminal problem.

#### 3.4.2 Implicit Addressing Feature of FreeBee

This section discusses the unique addressing scheme used in FreeBee. As a reminder, demodulating each interval-multiplexed FreeBee signals remains the same as the case for a single signal; folding with  $P$  yields the corresponding FreeBee signal with  $P$ .

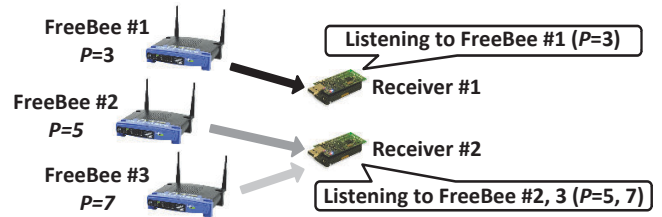


Figure 13: Implicit addressing via interval multiplexing

- **For stationary deployment:** As shown in Figure 13, each interval  $P$  is allocated to one freeBee sender, implicitly addressing that sender. Hence a receiver may select a subset of  $P$ 's that corresponds to the sender(s) of interest.
- **For mobile deployment:** In mobile scenarios, mobile devices are not aware of  $P$ s used by nearby FreeBee enabled devices. We need to fold for all  $P$ s in the prime set to receive from every sender. This is not as computationally heavy as it may sound, as the number of primes, by the prime number theorem [21], is limited to  $\frac{x_{max}}{\ln(x_{max})} - \frac{x_{min}}{\ln(x_{min})}$  when  $x_{max}$  and  $x_{min}$  are the maximum and minimum values in the prime set. For example, there are 20 primes in the interval range of  $x_{min} = 53$  and  $x_{max} = 149$ .

From a practical point of view, we emphasize that implicit addressing is a unique and effective feature of our design: as each sender is required to select different beacon interval, symbols demodulated with the same  $P$  are ensured to be from the same sender. This allows safely constructing a long symbol by appending received symbols. However, this is not the case for all other cross-technology techniques. In Esense [14] and HoWiES [41], a sender ID has to be embedded along with information in order to concatenate separate symbols correctly, leading to a large overhead in such a low-rate cross-technology communication.

## 4. PRACTICAL ISSUES

In this section, we discuss practical issues and their effects on our design.

- **Channel access delay:** Although beacons are prioritized over data packets and hence queueing delays are negligible [35], they *do* suffer from channel access delays according to CSMA. This is in fact a challenge uniquely imposed on our design, which is different from traditional the PPM environment where all pulses are transmitted at their exact times. Upon long delays, a beacon may fail to contribute to folding. This is precisely why beacon repetitions are needed (e.g., four beacons in Figure 7) for statistical performance guarantee. Our empirical study in Section 6.3 suggests that 5 beacons yield an error of less than 1%.

- **Noise:** Any non-beacon signal occupying the spectrum serves as noise and is a potential source of error. That is, as frequent 1's due to noise fill up the sampled RSSI, there is an increased chance of a large fold sum formed elsewhere to the beacon position, thus inducing demodulation failure. In other words, the performance of our design is enhanced by reducing the noise. This is simply done by taking only the first two RSSI samples for *any* packet including beacons and discarding (i.e., set to 0) the rest. The reason behind this approach is two-fold: (i) As data packets tend to be much longer than beacons, this reduces noise to 1/6 on average in our experiments. (ii) Our empirical analysis indicates that the channel access delays of beacons are mostly ( $>90\%$ ) less than 256 $\mu$ s, where a similar result was reported in a recent study [19]. Noting that the duration of a RSSI sample is 128 $\mu$ s, this suggests the first two RSSI samples (256 $\mu$ s) of beacons maintain a high chance of overlapping (i.e., contributing to fold sum) upon folding.

## 5. ANALYTICS

In this section, we provide a theoretical analysis of the performance of FreeBee and A-FreeBee.

### 5.1 SER vs. Sampling Duration

As noted earlier, the symbol error occurs upon demodulation failure, essentially when the fold sum of noise is higher than that of the beacons. For brevity, we defer a detailed SER (Symbol Error Rate) derivation to Appendix 9 and here move directly to the results to demonstrate the impact of three system parameters: (i)  $T$ , the beacon interval. (i)  $\rho$ , the number of beacon repetitions for statistical demodulation; and (iii) Sampling duration, the sampling time to obtain a symbol, which is  $\rho \times T$  for FreeBee and  $\rho \times 2T$  for A-FreeBee.

Figures 14(a) and 14(b) show Symbol Error Rate (SER) when default beacon interval is set as  $T = 97\Delta = 99.3ms$ .

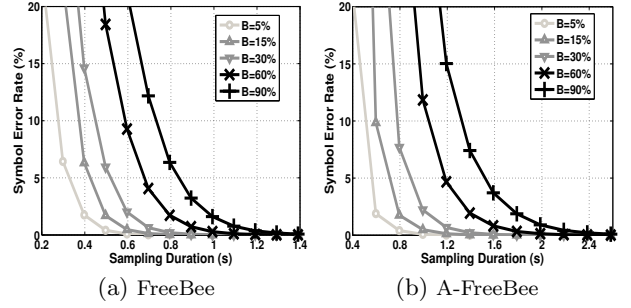


Figure 14: SER vs. sampling duration ( $T = 97\Delta$ ).

These figures convey three ideas: (i) longer sampling (i.e., higher  $\rho$ ) lowers SER, as more beacons are utilized to fight against noise; (ii) for a given duration, FreeBee achieves a lower SER than A-FreeBee; and (iii) higher channel occupancy, denoted by  $B$ , indicates more noise, thus higher SER. The figures have  $B$  up to 90% for completeness of analysis, where  $B \leq 30\%$  was observed in our experiment (under the threshold of -75dBm) in a university building with 50+ APs. Similar observations were reported in a recent study from a large-scale open WiFi trace [3, 43]. The figures suggest analytically that FreeBee and A-FreeBee achieve  $SER < 1\%$  for durations of 0.7s and 1.2s under such condition. We note this analytical result matches well with results from empirical experiments shown in Section 6.3.

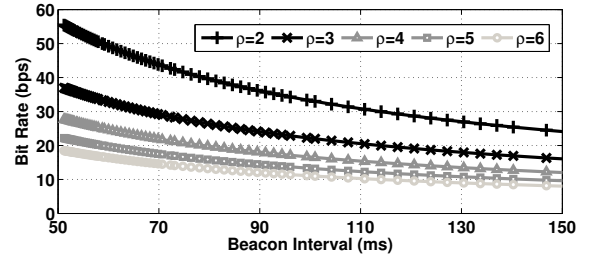


Figure 15: Bit rate vs.  $T$  for different  $\rho$ .

### 5.2 Transmission Rate vs. Beacon Interval

Beacon interval,  $T$ , is another factor that can affect (A-)FreeBee performance. The impact of beacon interval  $T$  can be observed with bit rate  $R$ . Intuitively, enlarging the beacon interval has two effects: (i) it offers more space for shift, or equivalently, yields more bits per symbol; and (ii) it requires more time to reach the same  $\rho$ . The bit rate for FreeBee can be computed as below:

$$R = \frac{\log_2 T / \Delta}{T \times \rho} \text{ bps} \quad (1)$$

Noting that  $\Delta$  (1.024ms in WiFi) defines the granularity of shift, the numerator in the Equ. 1 implies *bit per symbol*. Figure 15 shows the impact of beacon interval  $T$  on  $R$  in different scenarios for the range of practical intervals. In A-FreeBee, the rate is cut in half as it takes double sample duration (i.e.,  $\rho \times 2T$ ) to convey a same symbol. It is important to note that the rate given here is per sender, without bandwidth consumption (i.e., without incurring extra traffic). Due to interval multiplexing, the aggregated throughput linearly increases according to the number of senders. Furthermore, boosting the throughput of a single

sender by adopting additional beacons is also a viable option. Performances under such cases are shown via experimental evaluations in the next section.

## 6. PERFORMANCE EVALUATION

We show the generality of (A-)FreeBee by evaluations performed across four platforms operating on three different wireless standards: WiFi, ZigBee, and Bluetooth.

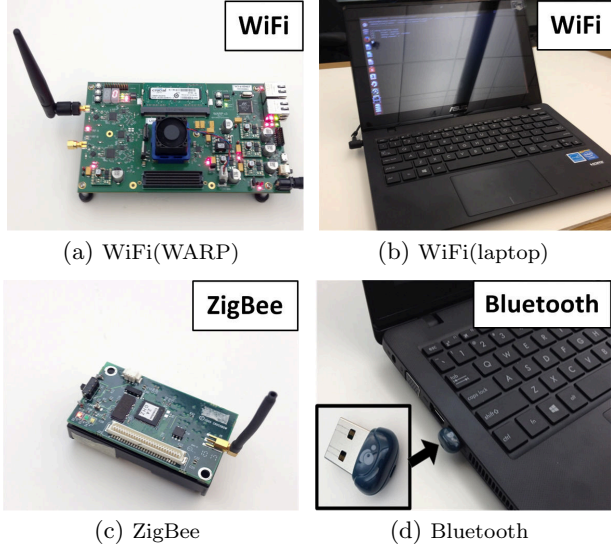


Figure 16: Four evaluation platforms on three different wireless technologies – WiFi, ZigBee, and Bluetooth.

Table 1: Experiment Settings

Communication Direction	Tx Ch.	Tx Power	Rx Ch.	Dist.
WiFi → ZigBee	1,4	13 dBm	11-15	8m
ZigBee → WiFi	11-14	0 dBm	1	1.5m
Bluetooth → WiFi	37-39	4 dBm	4	3m
Bluetooth → ZigBee	37-39	4 dBm	15	3m

### 6.1 Experiment Settings

The experiment parameters are specified in Table 1, where the detailed settings are as follows.

- **Design generality:** As a generic cross-technology communication framework, our design mechanisms including modulation, demodulation, and interval multiplexing commonly apply to different underlying technologies. This is possible because, according to the standards, (i) beacons are adopted in all three technologies of WiFi, ZigBee, and Bluetooth, (ii) they commonly allow modification of beacon timings, (iii) they reside on overlapping frequencies in the 2.4GHz band and finally, (iv) the light-weight design makes our design feasible even to low-end devices, as demonstrated later in the section.

- **WiFi:** Figures 16(a) and 16(b) show the two WiFi platforms on which our design is implemented: WARP [9] and laptops. The former is a wireless research platform fully integrated with WiFi PHY/MAC layers. As a FPGA-based system allowing real-time operation, the evaluation on WARP enables precise exploration into FreeBee performance. Further implementation on six different laptops with various

WiFi NICs from Qualcomm, Realtek, and Intel avoids hardware bias.

Evaluations via laptops utilize `Lorcon2` packet injection library [7] to schedule beacons according to FreeBee, which is a reasonable approach since laptops/PCs running software AP are frequently used in practice [4]. In Table 1, we use WiFi channel 1 (unless otherwise specified) for communication with ZigBee, one of the three most commonly used channels (i.e., 1, 6, and 11), granting generality to our result. We then tune to channel 4, which overlaps with a Bluetooth advertising channel (i.e., 38), for communication with Bluetooth.

- **ZigBee:** We use 30 ZigBee-compliant MICAz nodes (Figure 16(c)) for our experiments. When operating as a receiver, a MICAz node captures RSSI values (at 7.8KHz sampling rate by default) and records them within its 512KB on-board flash. The values are either processed (i.e., demodulated) within the node or flushed to a PC for analysis, depending on experiments. We use channels 11-15, overlapping with WiFi channels 1 and 4, and a Bluetooth advertising channel of 38.

- **Bluetooth:** Our design is implemented on IOGEAR Bluetooth LE USB adapter, a cheap (~12 USD) off-the-shelf product, shown in Figure 16(d). On this device, we utilize `AltBeacon` [1] library running under Linux’s BlueZ driver for FreeBee embedding. Specifically, connectable directed advertising was used to generate FreeBee-enabled beacons on all three advertising channels of 37-39, which complies to the standard on Bluetooth beaconing.

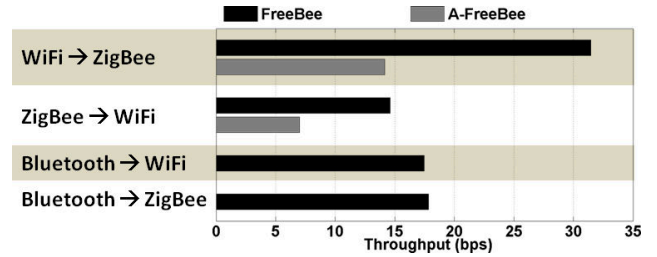
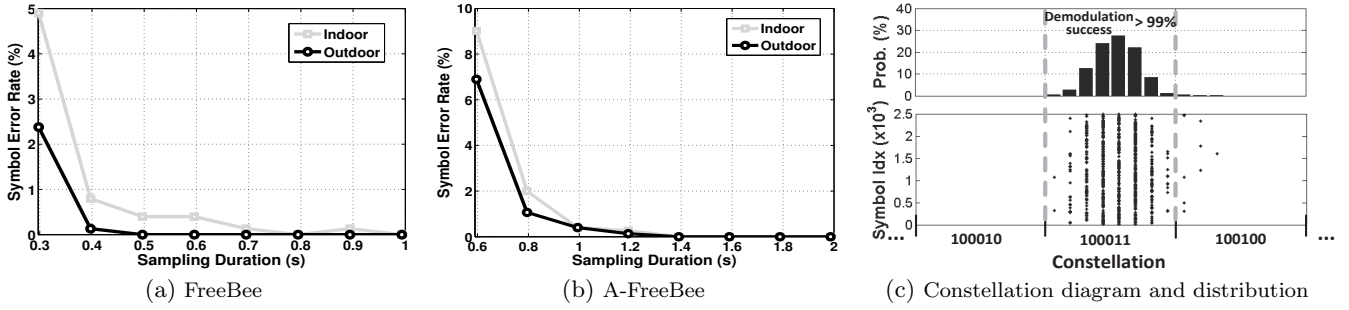


Figure 17: Per-sender throughputs achievable without incurring any extra traffic, evaluated on four different communication channels

### 6.2 Per-sender Throughput

In this section we demonstrate the data rate achievable per sender, under free side-channel as well as when the channel is fully utilized.

- **Free side-channel throughput:** Figure 17 illustrates per-sender data rate without introducing additional traffic (i.e., free side-channel). The experiment was conducted in a residential area with 20+ APs in proximity. Beacon intervals are set as 99.3, 76.8, and 78.75ms for WiFi, ZigBee, and Bluetooth, respectively, where the rate may easily be enhanced with shorter beacon intervals. The figure demonstrates two ideas: (i) The performance of A-FreeBee is slightly less than half of that of FreeBee, due to doubled sampling duration in A-FreeBee. Longer sampling duration increases the chance of larger fold sum of noise, hence yields higher SER in A-FreeBee compared to FreeBee. This agrees with our theoretical analysis in Section 5.1. (ii) Among different communication directions, WiFi to ZigBee exhibits the fastest rate of 31.5bps for FreeBee. The rate drops to



**Figure 18: Indoor and outdoor performances of (a)FreeBee and (b)A-FreeBee when  $T = 97\Delta = 99.3\text{ms}$ . Due to less noise outdoors, both FreeBee and A-FreeBee reach  $\text{SER} < 1\%$  at  $\rho = 4$ , whereas indoor environment required  $\rho = 5$ . (c) shows where the received symbols are positioned when ‘100011’ was sent 2,500 times.**

14.6bps for ZigBee to WiFi as the ZigBee standard enforces large unit shift (i.e.,  $\Delta = 15.36\text{ms}$ ), reducing the amount of information embedded in a symbol. While the Bluetooth standard defines a fine-grained beacon shift unit ( $\Delta = 0.625\text{ms}$ ), the random backoff ranging up to 10ms affects the performances of Bluetooth to WiFi and ZigBee communications where they show 17.5 and 17.8bps, respectively. While disabling the backoff functionality would increase the throughput significantly, the case is not considered in this paper as it requires modification to the standard (lacks compatibility).

• **Upper bound throughput under ideal conditions:** We demonstrate the effectiveness of our design by comparing the maximum throughput of our design to that of Esense [14], a state-of-art cross-technology communication scheme. Esense, as a WiFi to ZigBee communication technique, conveys data by modulating WiFi packet durations, where its ideal maximum throughput is reported under the ideal condition of interference-free channel that is fully utilized by a single sender. Hence we adopt this setting in evaluating FreeBee. Evaluation parameters also follow the values proposed in Esense; That is, RSSI sampling rate of  $32\text{KHz}$  ( $=$  sampling interval of  $30.5\mu\text{s}$ ), inter-frame duration of  $90\mu\text{s}$ , and the maximum WiFi transmission rate of  $54\text{Mbps}$  ( $802.11\text{g}$ ).

To obtain the ideal FreeBee performance, we first note that beacons need not be repeatedly transmitted under the interference-free channel. That is, a single beacon frame conveys a symbol where  $\text{SER} = 0$ . When the maximum shift is  $x\Delta$ , a symbol (i.e., a beacon) embeds  $\log_2(x+1)\text{bits}$ , where the time it consumes consists of inter-frame duration, beacon transmission time, and the amount of shift, where the unit of shift may be as small as the sampling interval (i.e.,  $\Delta = 30.5\mu\text{s}$ ). When  $x = 4$ , FreeBee yields the throughput of  $10.2\text{Kbps}$  given the beacon length of  $100\text{Bytes}$ . Esense, according to its researchers, achieves the throughput of  $5.13\text{Kbps}$  under the same setting. This is because Esense requires to use long-length packets (with long air-time) up to  $1,500\text{Bytes}$  to enable measurement of WiFi packet durations via a low-end ZigBee node. Meanwhile, FreeBee utilizes short beacons to offer higher channel efficiency.

### 6.3 Symbol Error Rate

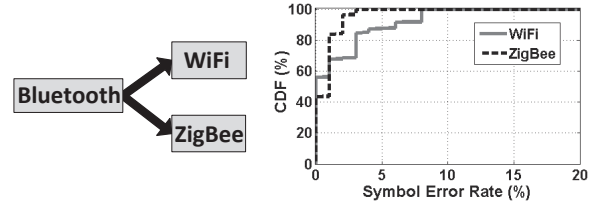
Here we present the reliability of our design in practice by evaluating SER under both indoor and outdoor environments in a residential area. This experiment is based on WiFi(WARP) to ZigBee communication, where more than 2,500 symbols are sent and demodulated for SER computation. The results for FreeBee and A-FreeBee are depicted in

Figures 18(a) and 18(b). Both designs reach  $\text{SER} \leq 0.5\%$  when  $\rho = 5$  (i.e.,  $0.5\text{s}$  and  $1\text{s}$  for FreeBee and A-FreeBee), regardless of the environment. Furthermore, both designs perform better in outdoor environments, due to lower channel occupancy,  $B$ .

The lower figure in 18(c) illustrates the constellation, along with the demodulated positions of received symbols, when the 6bit symbol of ‘100011’ is repeatedly sent 2,500 times. Demodulation is successful when a dot resides inside the region marked by gray dotted lines in the constellation. The upper figure in 18(c) shows the distribution of the dots, in which  $>99\%$  are successfully demodulated.

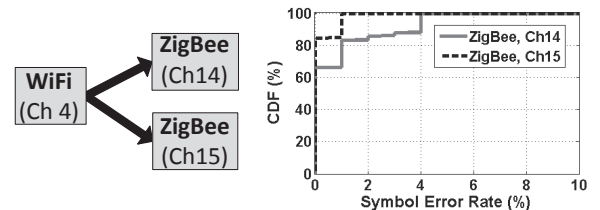
### 6.4 Cross-technology/channel Broadcast

This section demonstrates FreeBee’s unique capability to broadcast to receivers with heterogeneous technologies and channels.



**Figure 19: Simultaneous broadcast from Bluetooth to WiFi and ZigBee.**

• **Cross-technology broadcast:** Depicted in Figure 19, as a generic communication framework, FreeBee allows broadcast to heterogeneous receivers with overlapping frequencies. Our generic design is effective in practice, since it avoids the complexity associated with technology-specific operation. In the experiment, WiFi and ZigBee were set to channels 4 and 15 to listen to Bluetooth’s advertisement channel of 38 simultaneously. In this particular case shown in Figure 19, WiFi receiver, compared to ZigBee, suffers from larger SER. It is due to higher noise, as WiFi channel 4 overlaps with the popular WiFi channels of 1 and 6, while ZigBee channel 15 does not.



**Figure 20: Simultaneous broadcast from WiFi to ZigBee on different channels.**



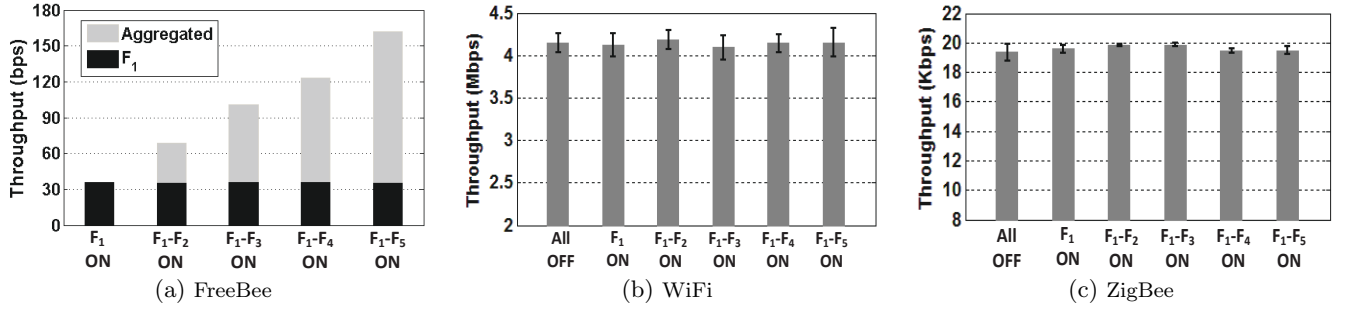


Figure 21: Transparency of FreeBee to (a) concurrent FreeBee signals, (b) WiFi, and (c) ZigBee

- **Cross-channel broadcast:** Figure 20 demonstrates another FreeBee's feature where a sender with wider bandwidth (WiFi) reaches multiple narrower-band receivers (ZigBee) on different channels with a single broadcast. WiFi's bandwidth spans 20MHz while it is only 2MHz for ZigBee. The experiment result in Figure 20 shows that SER on channel 14 is larger than that of channel 15. This is because channel 14 is affected by the noise from the heavily-used WiFi channel 1.

We note cross-technology broadcast and cross-channel broadcast can be combined to support more sophisticated scenarios where multiple heterogeneous senders deliver control symbols (through interval multiplexing) to multiple heterogeneous wireless receivers running under different channels. Such a capability would encourage further research on cross-technology coordination and control.

## 6.5 Transparency

We deploy five WiFi FreeBee senders,  $F_1$ - $F_5$ , with beacon intervals of  $\{91.1, 99.3, 103.4, 105.5, 109.6\}$  ms corresponding to  $\{89\Delta, 97\Delta, 101\Delta, 103\Delta, 107\Delta\}$ . Prime intervals indicate they are interval-multiplexed. Each sender is allowed to operate in FreeBee ON or OFF modes. A sender embeds FreeBee symbols only when it is ON. When OFF, the sender acts as a legacy AP to simply transmit beacons periodically without symbol embedding. Under this setting, we observe the throughput of legacy wireless networks when a single to multiple FreeBee APs are operational, as well as the cross-interference among them. To do so, we first turn only  $F_1$  to ON mode, while leaving all the others OFF. Then, we turn  $F_2 - F_5$  to ON one by one every 10 minutes.

- **Transparency between FreeBees:** As the black bar in Figure 21(a) demonstrates, the throughput of  $F_1$  is kept stable at an average of 35.5bps in the face of multiple concurrent FreeBee transmissions. This validates the performance of interval multiplexing in suppressing cross-interference among FreeBee signals. In fact simultaneous transmission helps to linearly increase the aggregated throughput up to 161.8bps, as shown by the gray bar. While the throughput can be enhanced by selecting shorter intervals for  $F_1 - F_5$ , we select ones close to that of a legacy AP ( $102.4ms = 100\Delta$ ) to limit the channel usage to a similar level.

- **Transparency to legacy networks:** We again turn  $F_1 - F_5$  to ON mode one at a time every 10 minutes, during which throughput between a pair of WiFi or ZigBee nodes are measured. The results are demonstrated in Figures 21(b) and 21(c). For WiFi, we use `Iperf` [6] to measure TCP throughput when operating in 802.11g. The WiFi sender was placed beside the FreeBees, 8m away from the WiFi receiver. For ZigBee measurement, sender and receiver were

placed 5m apart, where 20byte packets were continuously transmitted at the fastest speed (i.e., with the minimum inter-packet delay). The figures show stable throughputs averaging 4.1Mbps and 19.6Kbps for WiFi and ZigBee, respectively, with small standard deviations. This suggests that both networks are unaffected by the presence of FreeBees, confirming that the free side-channel design successfully keeps FreeBee transparent to legacy networks.

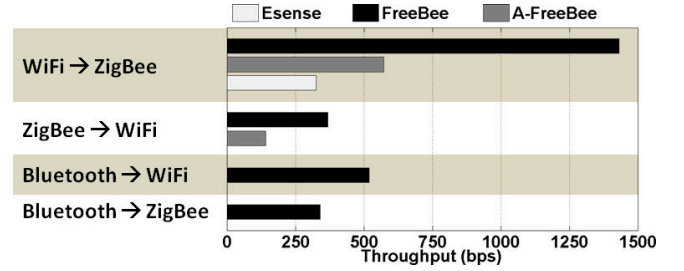
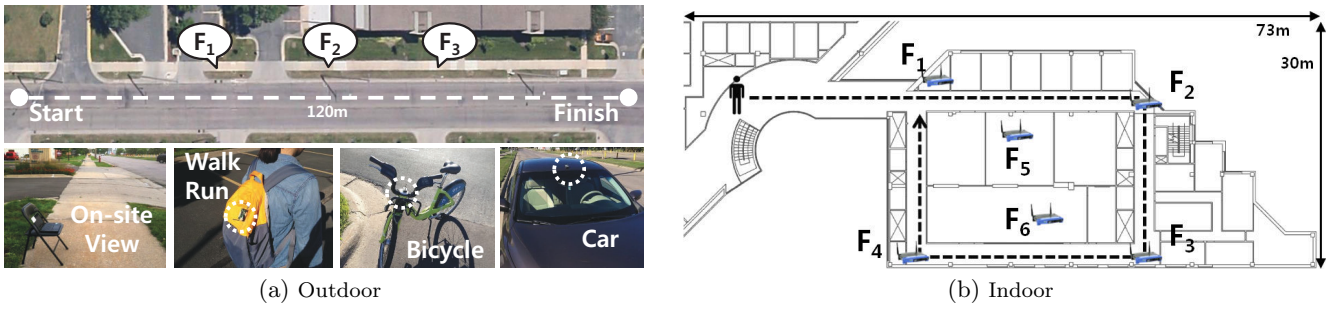


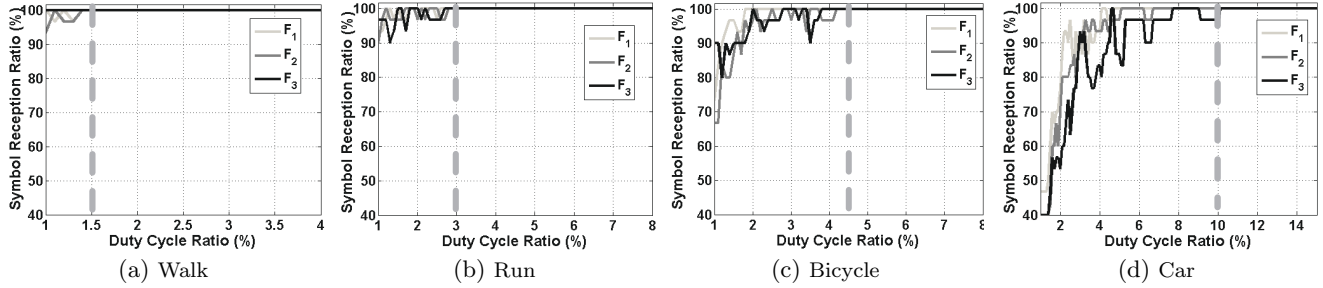
Figure 22: Aggregated throughput via interval multiplexing across multiple senders and/or within a sender. Compared to the state-of-the-art, Esense, FreeBee and A-FreeBee shows  $4.3\times$  and  $1.8\times$  performance, respectively.

## 6.6 Aggregated Throughput

This section investigates throughputs achievable via interval multiplexing, under different communication scenarios. Through interval multiplexing our design supports: (i) boosted aggregated throughput via concurrent transmission from multiple senders and (ii) adoption of additional beacons in a single sender to increase the throughput of that sender. We note that (i) and (ii) can be applied simultaneously; that is, multiple senders transmit concurrently while each sender freely adopts additional beacons in case of a need for higher throughput. Figure 22 depicts the throughputs for different communication scenarios, where we compare our design to the state of the art, Esense. According to the authors, when multiple Esense senders are present and the interference from the heterogeneous wireless systems are negligible, Esense achieves the bit rate of 1.63Kbps when a single WiFi packet is used to deliver one symbol, where five consecutive packets are needed per symbol for reliable communication [14]. This yields a throughput of 326bps. Following the same setting, Figure 22 demonstrates the rate achievable when the channel is shared among multiple (A-)FreeBee senders. FreeBee from WiFi to ZigBee shows the highest throughput of 1.4Kbps. The rate drops in other scenarios due to the same reasons discussed in the previous section; that is, a large shift unit in ZigBee and random ac-



**Figure 23: Mobile scenario evaluation:** we experimentally reveal the impact of mobility on FreeBee on both indoor and outdoor scenarios, when the receivers are duty-cycled.



**Figure 24: When walking, running, riding bicycle and car, reception can be ensured with duty cycling of only 1.5%, 3%, 4.5% and 10%, respectively.**

cess delay in Bluetooth. It is notable that Bluetooth to WiFi shows higher throughput of 514bps, compared to 349bps in the Bluetooth to ZigBee case. This is because higher sampling rate achievable in WiFi compared to ZigBee (10MHz versus 7.8KHz in our setting) allows precise measurement of the beacon timings and reduces the chance of noise forming a high fold sum.

We note that the results in Figure 22 is achievable under a large number ( $\sim$ hundreds) of concurrent senders, which can be realized by the following two techniques: (i) To allow numerous interval-multiplexed symbols, we choose beacon intervals from a set of numbers whose pair-wise LCM is longer than the length of the sampled RSSI vector. This clearly offers many more adoptable intervals compared to just using primes, while maintaining the effect of the interval multiplexing. (ii) As shown in the previous section, interval-multiplexed FreeBee symbols are transparent to each other under usual volume of concurrent senders (e.g., tens of FreeBee WiFi APs). Here we consider the extreme case of hundreds of senders where we take cross-interference into account. We address this issue of cross-interference by eliminating the beacons from the RSSI vector after they are interpreted. Specifically, we demodulate symbols in the ascending order according to their intervals. After demodulating each symbol, the beacons (i.e., RSSIs) corresponding to that symbol are eliminated (i.e., flipped from 1 to 0 in the RSSI vector). By doing so the cross-interference between symbols are effectively canceled to yield a high throughput. For instance, in WiFi to ZigBee FreeBee, SER is suppressed to be less than 5% when over 400 symbols with intervals between 91.1ms and 2s are concurrently sent.

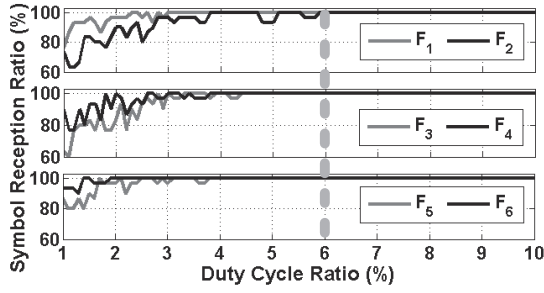
## 6.7 Mobility and Duty Cycling

Here we consider a practical scenario of mobile receivers. For instance, this insight offers ideas on the feasibility of

A-FreeBee for applications that requires location tracking. Furthermore, we take duty cycle into account where radios are turned off the majority of the time to preserve energy, a technique often adopted in battery-powered ZigBee networks to support long-term operations [11, 32, 24]. We deploy several WiFi A-FreeBee senders (laptop) in a university building and on a street, where the mobile receivers pass by the senders at different speeds: walking, running, on a bicycle, and in a car. It is worth noting that this experiment takes advantage of our free-side channel design, as (A-)FreeBee symbols can be continuously broadcast without occupying the channel, thus reaching mobile and/or duty-cycled receivers whose presence or active periods are unknown a priori.

- **On a street:** In this experiment, we deploy 3 A-FreeBee senders, 20m apart, on a street as shown in Figure 23(a). The figure also shows ZigBee receivers with different degrees of mobility: walking (4.3 mph), running (6.8 mph), on a bicycle (10.8 mph) and in a car (30 mph). Thirty receivers were tested for each speed. Figure 24 informs that A-FreeBee symbols from all the senders can be safely received with duty-cycling of only 1.5%, 3%, 4.5% and 10% when walking, running, cycling and riding in a car, respectively.

- **In a university building:** A total of 6 FreeBee senders are deployed as shown in Figure 23(b) to show the impact of various positions, such as hallways and rooms with closed or open doors. A person walks at a moderate speed of 4.3 mph through the hallway following the arrow in the figure, with a ZigBee receiver tied to his/her backpack. The result after repeating for 30 different receivers is shown in Figure 25. Overall, as indicated by the gray dotted line, duty cycling of 6% ensures reception of A-FreeBee symbols from all the senders. A higher duty cycle ratio is required compared to



**Figure 25: Duty cycling of 6% guarantees reception when walking in a university building.**

the 1.5% when walking outside in the previous experiment due to the noisy environment with 50+ legacy WiFi APs deployed.

## 6.8 Receiver Overhead

In this section we demonstrate light-weight storage and computation/energy overheads of (A-)FreeBee demodulation, making it affordable even to low-end devices (e.g., ZigBee mote) without disrupting the ongoing legacy protocols.

- **Storage:** Let us consider an example case where the sender is a WiFi AP with the beacon interval of  $T = 97\Delta = 99.3ms$  (i.e.,  $\lambda=776$ ) and the receiver is a ZigBee mote. As explained earlier, the receiver samples RSSI every  $128us$ , where the total number of  $\rho \times \lambda$  samples are collected for demodulation. As a RSSI value is represented as a single bit, under  $\rho = 5$  FreeBee requires  $5 \times 776 = 3.88Kbits$  (i.e., 485Bytes) of memory. This takes up less than 1% of the storage offered in popular commodity ZigBee motes of MICAz and TelosB, where they offer 512KBytes and 1MBytes of on-board flash, respectively. Moreover, it is also important to note the followings: First, concurrent demodulation of  $n$  interval-multiplexed FreeBee symbols whose intervals are  $\lambda_1 < \lambda_2 < \dots < \lambda_n$ , is achieved by collecting  $\rho \times \lambda_n$  samples. Second, it is possible to reduce the memory usage by storing only the timing of transitions (i.e.,  $0 \leftrightarrow 1$ ) in RSSI values. Third, A-FreeBee requires to store twice the number of samples compared to the basic FreeBee.

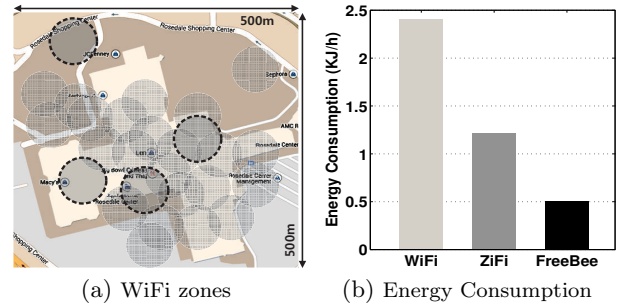
- **Computation/Energy:** The demodulation process can be decomposed into three parts: (i) sampling RSSI, (ii) adding to fold sum, and lastly, (iii) locating the position of the maximum fold sum. (i) and (ii) are repeatedly performed every  $128us$ , periodically incurring small overheads to perform a fetch and an addition operations. Upon collecting sufficient number (i.e.,  $\rho \times \lambda$ ) of samples (iii) takes place in which series of compare operations are executed in search for the maximum of the  $\lambda$  fold sums ( $2\lambda$  for A-FreeBee).

We demonstrate the overhead of the entire steps (i.e., (i)-(iii)) on a off-the-shelf, low-end system – ZigBee-compliant MICAz node. Specifically, we observe the execution time the node spends to process the demodulation. This is accurately measured by triggering a GPIO (general-purpose input/output) pin, whose activity is captured with Tektronix DPO 4054 oscilloscope. It is found that the demodulation of a symbol with an interval of  $\lambda = 776$  costs  $51.5ms$  of computation time, consuming only  $1.2mJ$  of additional energy compared to sleeping. The light overhead allows FreeBee to run on commodity low-end hardware without disrupting the legacy protocols. Finally, we note that demodulat-

ing multiple, interval-multiplexed FreeBee symbols requires more computation/energy where they still remain affordable; For instance, listening to 10 concurrent symbols costs  $98.5ms$  of computation, indicating  $2.4mJ$  of energy.

## 7. APP: WAKE ON SELECTIVE WLAN

In battery-powered mobile devices such as smart phones, WiFi NIC continuously scanning the channel for available APs, leading to significant energy drainage. To tackle this issue, recent studies utilize a low-power secondary radio to wake up the WiFi interface only when APs are present, since a WiFi interface consumes  $669.9mW$  [2] when it is awake, but it drops to only  $10mW$  once it starts to sleep. ZigBee, in contrast, requires only  $75mW$  [8], even when active. Among the works in this category, ZiFi [43] is a state-of-the-art solution that offers a simple, infrastructure-free approach to detect APs in proximity with a ZigBee SD card inserted in a smart phone.



**Figure 26: In (a), private and open WiFi zones are marked as light gray and dark gray (outlined) circles in a mall area; (b) shows the energy consumption, where FreeBee saves 78.9%.**

Although results in [43] are encouraging, we believe that further savings can be achieved with FreeBee. Given that most of APs nowadays are private, blindly waking up the WiFi without knowing if it is accessible or not leads to energy waste. This problem can be easily solved with FreeBee by allowing APs to broadcast binary information indicating if it is an open or a private WiFi. Figure 26(a) illustrates the benefit of this approach by considering a  $500 \times 500m^2$  area around a shopping mall. For a realistic evaluation, we observe all private and open WiFi APs currently running in the area<sup>1</sup>. Private WiFi APs are marked as light gray circles and open APs illustrated as outlined circles. The radius of the circles is set as  $43.7m$ , following the value obtained in our experiments. A total of 27 WiFi APs take up 43.2% of the entire area. On the other, hand only 4 open WiFi APs are found, covering only 9.6%.

To ensure a fair comparison, the energy consumption of the ZigBee node is also considered in case of ZiFi and FreeBee. The results in Figure 26(b) indicate that the energy consumption for WiFi, ZiFi, and FreeBee are  $2.41KJ/h$ ,  $1.21KJ/h$ , and  $0.51KJ/h$ , respectively. In sum, FreeBee saves 78.9% and 58.2% of the energy required by WiFi and ZiFi, respectively, without sacrificing opportunities for open WiFi access.

<sup>1</sup>Real AP deployment information retrieved from *wigle.net* and *openwifispots.com*.

## 8. RELATED WORKS

The FreeBee design of embedding information within the beacon timing is inspired by PPM (Pulse Position Modulation), which has been widely studied in the field of optical and UWB (Ultra Wide Band) communications [12, 15, 17, 27, 31, 37]. The extensive research in this area, however, is not applicable to our scenario because they are designed for pulses (which correspond to beacons in our case) that occur at their exact timing and/or can be differentiated from noise via a matched filter. Neither of these conditions is met by the beacons, since they are prone to channel access delays due to CSMA and they are indistinguishable from other traffic for receivers with incompatible PHY layers.

Improvement through cross-platform interaction has been investigated in many recent works [19, 22, 23, 26, 29, 30, 43], while only a handful have focused on cross-platform direct communication, including Esense [14], HoWiES [41], and GSense [40]. Esense establishes communication channels from WiFi to ZigBee by modulating packet lengths to those are unlikely to be used in normal WiFi traffic. HoWiES extends the Esense mechanism to convey data with combinations of WiFi packets, and introduces a simple coding technique. Finally, GSense enables cross-technology communication by prepending a customized preamble to legacy packets containing sequence of pulses in which gaps between them embed the data to be delivered. FreeBee uniquely departs from the above works in a few aspects: (i) the use of the mandatory beacons allows free-side-channel communication, (ii) interval multiplexing allows concurrent communication between multiple senders and receivers with implicit addressing, (iii) does not require modification to the commodity hardware (in contrast to GSense), and (iv) provides the first running prototype showcasing a generic communication capability among three popular wireless technologies.

## 9. CONCLUSION

In this paper we propose FreeBee, a cross-technology communication framework that aims to take advantage of the coexistence problem by adopting mutual supplementation. Extensive testbed experiments on three popular wireless technologies, WiFi, ZigBee, and Bluetooth, reveal that FreeBee offers reliable symbol delivery within less than a second. Its unique free side-channel design and the utilization of short beacon frames grant high spectrum efficiency and feasibility; FreeBee exhibits  $4.3\times$  the rate compared to the state-of-the-art, Esense, and demonstrates strong support for highly mobile (in a car) and extremely duty cycled ( $\leq 5\%$ ) receivers, implying its applicability to a wide range of practical applications. An examination of real WiFi deployment patterns in a shopping mall area finds that FreeBee can save 78.9% of the energy otherwise wasted by the WiFi interface.

## APPENDIX: SER DERIVATION

**Table 2: Parameters**

$T$	Beacon interval in time
$\lambda$	# of RSSI sample in $T$
$\rho$	# of repetitions
$B$	Channel occupancy rate
$B_f$	Channel occupancy rate after filter
$B_b$	Probability of beacon fold

From Table 2, recall that beacon repetition  $\rho$  is the number of rows in the matrix upon folding. The number of columns after folding is  $\lambda$  and  $2\lambda$  in FreeBee and A-FreeBee.  $B$  is the ratio of 1's in the sampled RSSI vector, reflecting the amount of traffic (i.e., noise) present in the channel.  $B_f$  stands for the ratio after taking only first two samples as introduced in Section 4. Due to such filtering, beacon is folded (i.e., contributes to fold sum) only when the delay of the beacon is less than  $256\mu s$ , where this probability is denoted as  $B_b$ .

• **FreeBee:** SER corresponds to the probability where fold sum of noise in *any* of  $\lambda - 1$  columns (excluding the one with beacon) is greater than or equal to that of beacon. The probability of a column to have  $n$  fold sum follows binomial distribution,  $\binom{\rho}{n} B_f^n (1 - B_f)^{\rho-n}$ . Let  $N$  denote a random variable indicating the *maximum* of the noise fold sums. Then,  $N$  is distributed as:

$$Pr(N \leq n) = \left\{ \sum_{i=0}^n \binom{\rho}{i} B_f^i (1 - B_f)^{\rho-i} \right\}^{\lambda-1}$$

$$Pr(N = n) = Pr(N \leq n) - Pr(N \leq n-1)$$

Letting  $S$  indicate the fold sum induced by beacons, then it is also binomially distributed:

$$Pr(S = s) = \binom{\rho}{s} B_b^s (1 - B_b)^{\rho-s} \quad (2)$$

where SER can be computed from  $S$  and  $N$ .

$$SER = Pr(S - N \leq 0)$$

$$= \sum_{s=0}^{\rho} \sum_{n=s}^{\rho} Pr(S = s) Pr(N = n)$$

• **A-FreeBee:** Symbol error occurs in A-FreeBee when fold sum of noise in any of  $2\lambda - 2$  columns (excluding the two with beacon) is greater than or equal to that of either beacon fold sums. Maximum noise fold sum,  $N$ , follows below distribution:

$$Pr(N \leq n) = \left\{ \sum_{i=0}^n \binom{\rho}{i} B_f^i (1 - B_f)^{\rho-i} \right\}^{2(\lambda-1)}$$

$$Pr(N = n) = Pr(N \leq n) - Pr(N \leq n-1)$$

The distribution of the beacon fold sum,  $S$ , is equivalent to Eq 2 where SER is found from  $S$  and  $N$ .

$$SER = 1 - \{Pr(S - N > 0)\}^2$$

$$= 1 - \left\{ \sum_{n=0}^{\rho-1} \sum_{s=n+1}^{\rho} Pr(S = s) Pr(N = n) \right\}^2$$

In summary, our analysis indicates that SER increases with  $B$  (or  $B_f$ ) and reduces with larger  $\rho$  as also shown in Figures 14(a) and 14(b).

## ACKNOWLEDGEMENT

This work was supported in part by the National Science Foundation under grants CNS-0845994 and CNS-1444021. We sincerely thank our anonymous shepherd and the reviewers for their valuable comments and feedback.



## 10. REFERENCES

- [1] Altbeacon. <http://altbeacon.org/>.
- [2] Cisco aironet 802.11a/b/g datasheet. <http://www.cisco.com>.
- [3] Crawdad: A community resource for archiving wireless data at dartmouth. <http://crawdad.cs.dartmouth.edu/>.
- [4] Hostapd. <http://hostap.epitest.fi/hostapd/>.
- [5] ios: Understanding ibeacon. <http://support.apple.com/kb/HT6048/>.
- [6] Iperf. <https://iperf.fr/>.
- [7] Lorcon wireless packet injection lib. <https://code.google.com/p/lorcon/>.
- [8] Micaz datasheet. <http://www.memsic.com/>.
- [9] Wireless open-access research platform. <http://warpproject.org/trac/>.
- [10] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In *Proc. ACM IMC*, 2009.
- [11] Z. Cao, Y. He, and Y. Liu. L 2: Lazy forwarding in low duty cycle wireless sensor networks. In *INFOCOM*, pages 1323–1331, 2012.
- [12] C. Carbonelli and U. Mengali. M-ppm noncoherent receivers for uwb applications. *Wireless Communications, IEEE Transactions on*, 5(8):2285–2294, 2006.
- [13] R. Carroll, R. Cnossen, M. Schnell, and D. Simons. Continua: An interoperable personal healthcare ecosystem. *Pervasive Computing, IEEE*, 6(4):90–94, 2007.
- [14] K. Chebrolu and A. Dhekne. Esense: communication through energy sensing. In *MOBICOM*, 2009.
- [15] A. A. D’Amico, U. Mengali, and E. Arias-de Reyna. Energy-detection uwb receivers with multiple energy measurements. *Wireless Communications, IEEE Transactions on*, 6(7):2652–2659, 2007.
- [16] N. Ding, D. Wagner, X. Chen, A. Pathak, Y. C. Hu, and A. Rice. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. In *ACM SIGMETRICS Performance Evaluation Review*, volume 41, pages 29–40. ACM, 2013.
- [17] Z. Ghassemloooy, A. Hayes, N. Seed, and E. Kaluarachchi. Digital pulse interval modulation for optical communications. *Communications Magazine, IEEE*, 36(12):95–99, 1998.
- [18] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the rf smog: making 802.11n robust to cross-technology interference. In *SIGCOMM*, 2011.
- [19] T. Hao, R. Zhou, G. Xing, and M. Mutka. Wizsync: Exploiting wi-fi infrastructure for clock synchronization in wireless sensor networks. In *RTSS*, 2011.
- [20] A. C. Hsu, D. S. L. Wei, and C. C. J. Kuo. Coexistence wi-fi MAC design for mitigating interference caused by collocated bluetooth. *IEEE Trans. Computers*, 64(2):342–352, 2015.
- [21] A. E. Ingham. *The Distribution of Prime Numbers*. Cambridge University Press, 1932.
- [22] T. Jin, G. Noubir, and B. Sheng. Wizi-cloud: Application-transparent dual zigbee-wifi radios for low power internet access. In *INFOCOM*, pages 1593–1601, 2011.
- [23] W. Li, Y. Zhu, and T. He. Wibee: Building wifi radio map with zigbee sensor networks. In *INFOCOM*, 2012.
- [24] Z. Li, M. Li, and Y. Liu. Towards energy-fairness in asynchronous duty-cycling sensor networks. *TOSN*, 10(3):38, 2014.
- [25] C.-J. M. Liang, B. Priyantha, J. Liu, and A. Terzis. Surviving wi-fi interference in low power zigbee networks. In *SenSys*, 2010.
- [26] R. Mahindra, H. Viswanathan, K. Sundaresan, M. Y. Arslan, and S. Rangarajan. A practical traffic management system for integrated lte-wifi networks. In *MOBICOM*, pages 189–200, 2014.
- [27] R. C. Qiu, H. Liu, and X. Shen. Ultra-wideband for multiple access communications. *Communications Magazine, IEEE*, 43(2):80–87, 2005.
- [28] B. Radunovic, R. Chandra, and D. Gunawardena. Weeble: enabling low-power nodes to coexist with high-power nodes in white space networks. In *CoNEXT*, 2011.
- [29] S. Sen, T. Zhang, M. M. Buddhikot, S. Banerjee, D. Samardzija, and S. Walker. A dual technology femto cell architecture for robust communication using whitespaces. In *Dynamic Spectrum Access Networks (DYSPAN)*, 2012.
- [30] C. Sengul, M. Bakht, A. F. Harris, T. Abdelzaher, and R. H. Kravets. On the feasibility of high-power radios in sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12(1):37–39, 2008.
- [31] D.-s. Shiu and J. M. Kahn. Differential pulse-position modulation for power-efficient optical communication. *Communications, IEEE Transactions on*, 47(8):1201–1210, 1999.
- [32] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and C. Petrioli. Beyond Duty Cycling: Wake-up Radio with Selective Awakenings for Long-lived Wireless Sensing Systems. In *INFOCOM*, 2015.
- [33] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An empirical study of low-power wireless. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):16, 2010.
- [34] D. Staelin. Fast folding algorithm for detection of periodic pulse trains. *Proceedings of the IEEE*, 57:724 – 725, 1969.
- [35] Wireless LAN Working Group. Ieee standard part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012.
- [36] Wireless Personal Area Network (WPAN) Working Group. Ieee standard part 15.4: Low-rate wireless personal area networks (lr-wpans). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pages 1–314, Sept 2011.
- [37] J. Wu, H. Xiang, and Z. Tian. Weighted noncoherent receivers for uwb ppm signals. *Communications Letters, IEEE*, 10(9):655–657, 2006.
- [38] X. Zhang and K. G. Shin. Enabling coexistence of heterogeneous wireless systems: case for zigbee and wifi. In *MobiHoc*, 2011.

- [39] X. Zhang and K. G. Shin. Cooperative carrier signaling: harmonizing coexisting wpan and wlan devices. *Networking, IEEE/ACM Transactions on*, 21(2):426–439, 2013.
- [40] X. Zhang and K. G. Shin. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *INFOCOM*, 2013.
- [41] Y. Zhang and Q. Li. Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices. In *INFOCOM*, 2013.
- [42] Z. Zhao, X. Wu, X. Zhang, J. Zhao, and X. Li. Zigbee vs wifi: Understanding issues and measuring performances of their coexistence. In *IPCCC*, pages 1–8, 2014.
- [43] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma. Zifi: wireless lan discovery via zigbee interference signatures. In *MOBICOM*, 2010.