Exploiting WiFi Guard Band for Safeguarded ZigBee

Yoon Chae
Department of Computer Science,
George Mason University
ychae2@gmu.edu

Shuai Wang
Department of Computer Science,
George Mason University
swang42@gmu.edu

Song Min Kim
Department of Computer Science,
George Mason University
song@gmu.edu

ABSTRACT

Cross-technology interference (CTI) from dense and prevalent wireless has become a primary threat to low-power IoT. This paper presents G-Bee, a CTI avoidance technique that uniquely places ZigBee packet on the *guard band* of ongoing WiFi traffic, which effectively *safeguards* the packet from WiFi interference. Such design ensures reliable ZigBee communication even under saturated WiFi traffic where traditional ZigBee is considered inoperable. Technical highlight is in lighweight WiFi guard band capture mechanism using ZigBee PHY layer samples directly accessible in various commercial ZigBee chip. Another exclusive feature of G-Bee is spectrum-synchronized low duty cycling – by utilizing guard bands of periodic WiFi beacons, active slots are effectively synchronized to spectrum availability (i.e., guard band) for significant delay improvement. Extensive evaluations on our prototype system demonstrates G-Bee PRR over 95% where legacy ZigBee drops to below 15% under significant interference with hundreds WiFi users and reduction of low duty cycle delay by 87.5%, all of which are achieved with a light computational overhead of 0.3%.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; • **Networks →** *Wireless personal area networks*;

KEYWORDS

Guard band, Cross-technology interference, ZigBee, WiFi

ACM Reference Format:
Yoon Chae, Shuai Wang, and Song Min Kim. 2018. Exploiting WiFi Guard Band for Safeguarded ZigBee. In *The 16th ACM Conference on Embedded Networked Sensor Systems (SenSys '18), November 4–7, 2018, Shenzhen, China.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3274783.3274835

1 INTRODUCTION

The massive body of wireless is anticipated to make another leap in the upcoming Internet of Things (IoT) era, up to 20 billion by 2020 [23, 25, 37]. While the dense coverage provides great convenience, this has led to severe competition over the shared ISM band. Specifically, cross-technology interference (CTI) [15] between heterogeneous devices has become one of the major bottlenecks to network performance. Studies have shown that the impact of CTI is especially significant for low-power IoT – WiFi interference is a major source of ZigBee disruption, corrupting over 50% of packets [4, 26, 29, 42].

Researchers have proposed a body of work on effectively avoiding CTI [2, 14, 16, 17, 32, 40, 44], where they commonly use statistical approaches to model the traffic pattern from which the channel idle times are inferred. For examples, WISE [17] predicts the idle time following the bursty WiFi while TIIM [16] adopts machine learning techniques to characterize CTI pattern. While shown to be highly effective, the *probablistic* nature of such designs leads to intrinsic vulnerability to dynamics of wireless activities, including the degree of traffic, number of users, and mobility.

We present G-Bee, a new technique that uniquely avoids CTI in a *deterministic* manner by counterintuitively exploiting ambient WiFi (802.11b) to *safeguard* ZigBee. This is achieved by leveraging the guard band of ongoing WiFi transmission, sufficiently wide (5 or 3MHz) to incorporate a ZigBee channel (2MHz); As originally intended to avoid inter-channel interference, WiFi guard band is designed such that it is ensured to be idle (i.e., free from WiFi interference). By carefully observing WiFi signal, G-Bee identifies guard band duration and frequency on the fly in which the ZigBee packet is placed, thus providing deterministic CTI avoidance.

The key contribution of G-Bee is ensuring reliable ZigBee under CTI, even under saturated WiFi traffic where ZigBee is normally considered inoperable. Our empirical evaluations demonstrate over 95% packet reception rate (PRR) where legacy ZigBee drops below 15%. G-Bee also exclusively enables spectrum-synchronized low duty cycling – i.e., by aligning the active slot with the guard band of periodic WiFi beacons, G-Bee essentially synchronizes active slot with the spectrum availability, offering low delay and high energy efficiency simultaneously. Other unique benefits include: (i) enhancing spectral efficiency by utilizing guard band which is otherwise wasted, (ii) non-disruptiveness to coexisting networks including WiFi, and finally (iii) backward compatibility with (i.e., transparently operates with) legacy ZigBee network, ensuring low adaptation cost.

Technical highlight of G-Bee is in lightweight detecting and decoding WiFi packet by smartly processing ZigBee PHY layer samples (in-phase and quadrature) directly accessible in various commercial ZigBee and other IoT RF, e.g., BLE [1, 6, 20, 30, 33]. The techniques are carefully designed to keep the computational/storage overhead and energy consumption minimal on off-the-shelf device. According to our measurement, G-Bee incurs only 0.3% computational overhead under feasible scenarios, demonstrating the practical pathway to utilizing PHY layer for IoT improvement. Our contributions are three-fold:

SenSys '18, November 4–7, 2018, Shenzhen, China

ACM ISBN 978-1-4503-5952-8/18/11…$15.00
https://doi.org/10.1145/3274783.3274835

- G-Bee uniquely investigates the opportunities of guard band for safeguarded communication in the ISM band, for intrinsic robustness to interference dynamics. While the case of ZigBee was examined, the idea can be generally applied to other IoT platforms, e.g., BLE.
- To the best of our knowledge, for the first time, we design and demonstrate lightweight practical techniques to process PHY layer samples from heterogeneous signal (WiFi ↔ ZigBee) for performance improvement.
- We implement G-Bee prototype on off-the-shelf design ZigBee testbed, where extensive real-life evaluations under hundreds of WiFi users yield PRR of 95%, improving the legacy ZigBee by a vast amount of 6.3×.

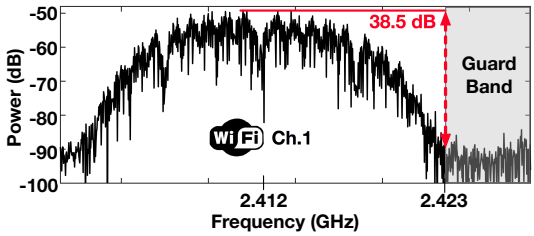


Figure 1: The guard band is conservatively set to 3 or 5MHz while WiFi devices effectively suppress leakage (38.5dB power drop) into the guard band, providing potential opportunity for exploitation.

2 MOTIVATION

Here we discuss the hidden opportunity behind the WiFi spectrum with potential to enable safe and spectrum-efficient ZigBee.

2.1 Opportunity Behind WiFi Guard Band

As per the standard, WiFi channels are separated by guard band. To avoid inter-channel interference, the guard band is set conservatively at 3 (802.11b) or 5MHz (802.11g/n) – larger than what is needed in most real-life scenarios especially with recent WiFi radio circuitry effectively suppressing signal leakage. Measurement in Figure 1 validates that WiFi signal from a commodity device (NETGEAR WNDR3800), emitted at the highest power of 20dBm, undergoes a large power drop of 38.5dB at the guard band. The measurement was taken from a USRP in close proximity (50cm), where the results were similar for other WiFi devices. This observation, along with the guard band width of 3 or 5MHz (>2MHz-wide ZigBee) demonstrates the potential of leveraging guard band to carry ZigBee. This improves spectral efficiency by recycling frequency (i.e., guard band) mostly left used.

More importantly, WiFi in fact *safeguards* (from interference) ZigBee within its guard band. This counterintuitive observation can be easily understood from the channel layout shown in Figure 2 – Suppose a WiFi device is transmitting on channel 1, which we represent[1] as Ch_1^w. Then, as per CSMA, other WiFi stations operating on overlapping frequencies (i.e., Ch_1^w-Ch_5^w) back off. This holds for all 802.11b, g, n and their combinations. In the meantime, Ch_6^w

[1]For clarity and brevity, throughout the paper we use notations Ch_i^w and Ch_j^z to indicate WiFi channel i and ZigBee channel j, respectively.

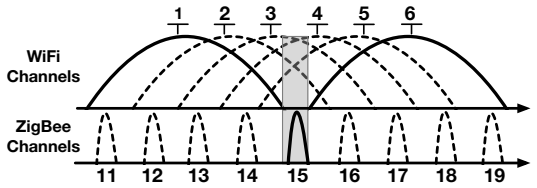


Figure 2: 5MHz-wide guard band (in gray) between two frequency-multiplexed WiFi channels of 1 and 6, with 2MHz bandwidth, fits within this under-utilized spectrum. Such guard bands exist adjacent to any WiFi channel.

may be used simultaneously as it does not overlap, where the guard band still remains idle. In other words, ongoing WiFi transmission on Ch_1^w ensures silence on the guard band, thus protecting ZigBee signal within that band. We note that ZigBee channel 15 (i.e., Ch_{15}^z) fits exactly within the guard band. Therefore, ZigBee on Ch_{15}^z is protected whenever WiFi is transmitting on Ch_1^w (or Ch_6^w). The same holds for WiFi on Ch_2^w and ZigBee on Ch_{16}^z. Utilizing guard band is indeed very powerful, as ZigBee is safeguarded *regardless* of the density of WiFi devices or their traffic volume, enabling ZigBee even under the worst interference scenarios.

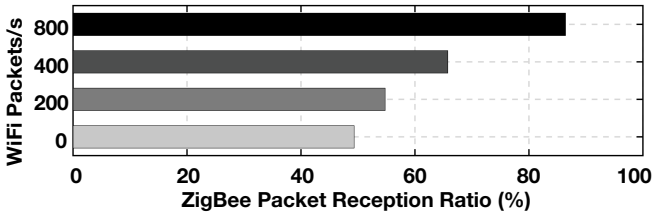


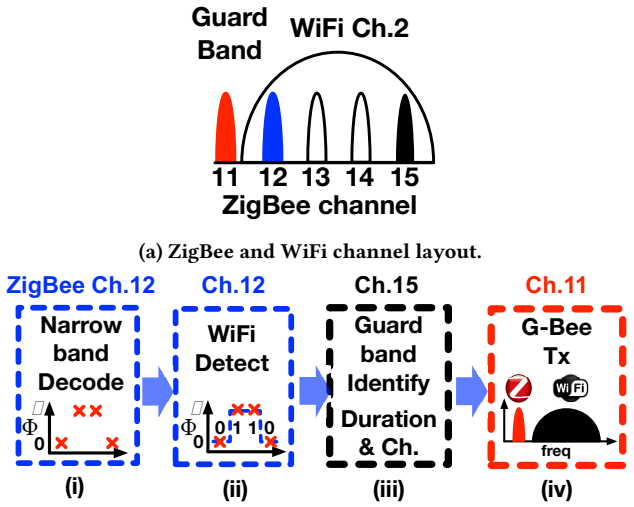
Figure 3: WiFi traffic (@Ch_1^w) vs ZigBee PRR (@Ch_{15}^z).

To validate such insight in practice, we deploy 5m-apart ZigBee pair exchanging packets under heavy WiFi. To examine the impact of guard band, we tune the ZigBee to operate on Ch_{15}^z and increase the degree of WiFi traffic on Ch_1^w with `Iperf`, which introduces more frequent guard band to protect more ZigBee transmissions. As depicted in Figure 3, 86.4% of 1,000 packets were successfully received under 800 pkts/s WiFi traffic on Ch_1^w, while it drops to only 49.3% without the WiFi traffic. The results remained consistent under various proximity and LOS or NLOS scenarios, demonstrating the efficacy of utilizing guard band for ZigBee under WiFi presence. The result contradicts the common belief that WiFi is only detrimental to ZigBee; in fact, WiFi may indeed protect ZigBee communications from WiFi interference via guard band.

Challenges in Practice. Although facilitation of guard band for ZigBee offers significant advantages in both spectrum and energy, it is non-trivial to achieve in reality. This is because guard band is highly dynamic in practice; That is, frequency, time, and duration of guard band reflects the channel, access time, and packet length/data rate/modulation of WiFi transmissions, which are unknown apriori. Specifically, since the channel access among WiFi devices are fully distributed and uncoordinated in nature, exploiting guard band imposes challenging tasks of real-time guard band capturing and utilization techniques that are generally applicable to any practical scenario, including traffic volume, pattern, and deployment density.

2.2 Feasibility of PHY Layer Utilization

Various recent commercial low-power radio chips from major vendors (e.g., Atmel and TI) offer direct accessibility to physical layer signal (In-phase and Quadrature samples) [1, 6, 30, 33–35], including ZigBee. This is mainly driven by ZigBee's limited bandwidth; Unlike wider band systems such as WiFi (>20MHz), ZigBee's bandwidth only spans 2MHz with a conventional sampling rate of 4MHz (vs. 40MHz or higher in WiFi [27]). With the modern low-power microprocessors equipped in ZigBee devices often reaching hundreds of MHz in speed [1, 31], the rate of 4MHz can be appropriately handled in software, especially with a careful (i.e., lightweight) design. For example, our design requires 13 cycles per 1us for computation and only 0.3% computational overhead for practical use. (Section 7).



(a) ZigBee and WiFi channel layout.

(b) G-Bee operation procedure.

Figure 4: Exploiting channel layout in (a), G-Bee identifies guard band duration/channel for exploitation via the steps in (b).

3 G-BEE OVERVIEW

G-Bee enables safeguarded ZigBee communication by leveraging guard bands of ambient WiFi, without any prior knowledge nor coordination. That is, G-Bee is entirely passive and highly flexible, where it accommodates any traffic pattern, volume, or density of WiFi in vicinity.

3.1 G-Bee Operation

G-Bee is essentially an add-on functionality at the ZigBee sender side, which detects the timing and duration of the guard band to transmit a ZigBee packet while the guard band is maintained (i.e., WiFi packet is on-air). The receiver side remains to be a standard ZigBee receiver. In other words, for a given ZigBee channel allocated between a sender-receiver pair, G-Bee captures the corresponding guard band (= detects WiFi packet in the corresponding channel) in real time, and transmits on the given channel accordingly.

In essence, G-Bee exploits the unique channel layout between ZigBee and WiFi, depicted in Figure 4a, where we leverage the fact that a WiFi channel can be specified by the overlapping leftmost and rightmost ZigBee channels. For example, if a WiFi packet is detected on (i.e., overlaps with) both Ch^z_{12} and Ch^z_{15}, the channel it resides on must be Ch^w_2. G-Bee not only captures the guard band of interest, but also its duration, by directly interpreting WiFi packet header.

As a walk-through example, suppose the ZigBee receiver is listening on Ch^z_{11}. From the channel layout, G-Bee (i.e., the sender) immediately finds that the ZigBee channel falls within the guard band of Ch^w_2. Then, G-Bee switches to Ch^z_{12}, the leftmost channel that overlaps with Ch^w_2 to capture WiFi packet. Subsequent processes are illustrated in Figure 4b – (i) G-Bee performs narrow decoding, (ii) where the decoded bits are matched to a fixed 38bit sequence for WiFi detection (Section 4.3). (iii) Once detected, G-Bee performs guard band identification (Section 4.4); i.e., G-Bee switches to Ch^z_{15}, the rightmost channel overlapping with Ch^w_2, where it continues to decode for WiFi packet length corresponding to guard band duration. Successful reading of length also indicates the WiFi packet is indeed in Ch^w_2. Finally, (iv) G-Bee switches to Ch^z_{11} to transmit for the duration of the guard band. This mechanism not only applies to Ch^z_{11}, but any ZigBee channels that overlaps with WiFi.

3.2 Leveraging 802.11b Guard Band

G-Bee exploits WiFi (802.11b) guard band with the following unique features for ZigBee protection: (i) it is free from WiFi interference including 802.11b, g, and n, (ii) long duration of often over 1ms (due to long 802.11b packets) suitable for protecting ZigBee packets, and (iii) periodically occurs thereby naturally supports ZigBee low duty cycling. This is because beacons are necessarily sent by WiFi APs typically every 102.4ms.

Ambient 802.11b Traffic. While 802.11g and n traffics are dominant, 802.11b is constantly utilized for its robustness, where it reaches the volume sufficient to support sparse IoT traffic. Specifically, majority of the recent WiFi devices are 802.11b/g/n compatible, where they often transmit control packets (e.g., beacons, probe request/response) in 802.11b for maximum reliability. According to our measurement study, more than 100 802.11b packets per second were found in a typical downtown environment (Sections 7.4). Therefore, G-Bee is widely applicable to practical WiFi settings and provides sufficient opportunities for IoT traffic.

3.3 Unique Features

G-Bee effectively cuts down the energy dissipation by 83% (Section 7.6), while enhancing the spectrum utilization by exploring largely under-utilized spectrum (i.e., guard band), improving networks as a whole. Also, G-Bee operates silently without disrupting WiFi, and is transparent to the ZigBee receiver. This offers great adaptability without enforcing replacement of existing network deployments. Finally, the idea of G-Bee applies to other IoT standards as long as it fits into the guard band and is capable of accessing physical layer information, e.g., BLE with commodity RF chips DA14781 [6] and SX1257 [30].

4 G-BEE DESIGN

This section illustrates the G-Bee design in detail.

4.1 Background

G-Bee exploits 802.11b WiFi guard band captured on the fly via ZigBee Rx front-end, which is achieved by the key technique of narrowband
layers of WiFi and ZigBee.

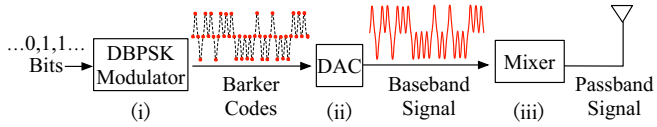


Figure 5: WiFi transmitter RF front-end.

WiFi Tx front-end. Figure 5 illustrates the architecture of WiFi transmitter [9]. (i) First, Differential Binary Phase-Shift Keying (DBPSK) mapper encodes the input bits to Barker code (non)inversions, which essentially stretches the bits to 1us-long sequences (i.e., Barker code) for reliability, generally known as Direct Sequence Spread Spectrum (DSSS). In (ii) the Barker code is filtered to yield band-limited (i.e., 22MHz) signal. This is then converted to analog continuous waveform denoted as $w(t)$ via digital-to-analog converter. In step (iii) $w(t)$ is shifted to passband by mixing with the carrier wave with the frequency corresponding to WiFi channel, f_w. This yields $w(t)e^{j2\pi f_w t}$, which is pushed into the air through the antenna.

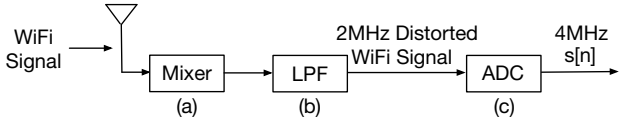


Figure 6: ZigBee receiver RF front-end.

ZigBee Rx front-end. As in Figure 6 WiFi signal of $w(t)e^{j2\pi f_w t}$ naturally flows into ZigBee radio coexisting on 2.4GHz band, which is then processed through steps (a)-(c). In (a) the signal is shifted to the baseband via mixing with ZigBee carrier signal at frequency f_z. The resulting signal is centered around the frequency difference between ZigBee and WiFi center frequencies (i.e., $f_w - f_z$). In (b) the signal passes through a low pass filter (LPF) that cuts off the 22MHz WiFi signal to 2MHz (i.e., ZigBee bandwidth) causing significant distortion. Lastly in (c) the signal is digitized by analog-to-digital converter (ADC) typically operating at 4MHz sampling rate. Then, n-th 4MHz sample $s[n]$ after passing through ZigBee mixer, LPF, and ADC becomes:

$$s[n] = (w(nT_s)e^{j2\pi(f_w - f_z)nT_s}) * h \quad (1)$$

where $*$ indicates convolution, while h is the time-invariant LPF with 2MHz bandwidth and T_s is the sampling interval of 250ns ($=\frac{1}{4MHz}$) in compliance with ZigBee.

4.2 Narrowband Decoding

G-Bee's key technique of narrowband decoding exclusively enables interpretation of WiFi bits, directly from ZigBee's physical layer (i.e., I/Q) samples. That is, counterintuitively, 22MHz bandwidth WiFi (802.11b) signal is decoded at the ZigBee RF front-end with only 2MHz bandwidth – thus narrowband decoding. In fact, the technique leverages unique signature in the WiFi signal distorted by 2MHz LPF at ZigBee, which can be generally applied regardless of carrier frequency offset (CFO) as well as WiFi and ZigBee channels.

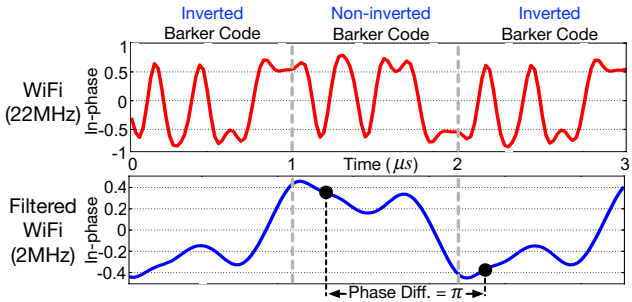


Figure 7: Upper figure demonstrates WiFi DBPSK using Barker code, where inverted and non-inverted codes are symmetric about the x axis. Lower figure shows that this feature is kept intact in the distorted signal after passing through 2MHz LPF; phase difference of π and 0 in 1μs interval indicates WiFi bits '1' and '0', respectively.

As seen in Figure 6, when under overlapping spectrum, WiFi DBPSK signal naturally flows into ZigBee RF front-end and passes through 2MHz LPF, leaving only low frequency components. Figure 7 depicts empirical measurements of the WiFi signal transmitted from a commercial WiFi (Intel 5300). The upper figure shows the original WiFi with 22MHz bandwidth, while the lower is the distorted signal after passing through 2MHz LPF, where both are measured on USRP for analysis purpose.

As in the upper part in Figure 7, WiFi DBPSK utilizes (non-)inverted Barker code where two consecutive (non-)inverted codes indicate bit '0' while the transition (inverted ↔ non-inverted) represents bit '1'. More importantly, we observe that the inverted and non-inverted Barker codes are symmetric about the x axis. As demonstrated in the lower figure, the feature is strictly retained even after the significant distortion induced by 2MHz filtering. Intuitively, this is because Barker code is essentially a 1μs-long pulse – therefore it may only repeat once per 1μs, or equivalently at 1MHz ($=\frac{1}{1\mu s}$), a frequency that can be captured within 2MHz. The feature of symmetry about the x axis is the fundamental enabler to narrowband decoding. For example, black dots in the lower figure illustrate the sign flip in the magnitudes 1μs apart (i.e., Barker code length) which reflects the code transition and thus represents bit '1'. On the contrary, the magnitude remains the same for consecutive (non-)inverted codes indicating bit '0'. In the complex domain, this translates to phase difference of π and 0 for bits '1' and '0', respectively (quadrature is 0). Therefore, narrowband decoding is computed as:

$$\Phi[n] = tan^{-1}s[4n] - tan^{-1}s[4(n-1)] \quad (2)$$

indicating $\Phi[n]$ is computed every $1\mu s$, or equivalently, every 4 samples (4MHz). To accommodate noise in reality, the decision boundaries equally divide the entire phase space of 2π centered around ideal constellation points of π (bit '1') and 0 (bit '0'): $\frac{\pi}{2} \leq \Phi[n] < \frac{3\pi}{2}$ is decoded as bit '1', otherwise bit '0'.

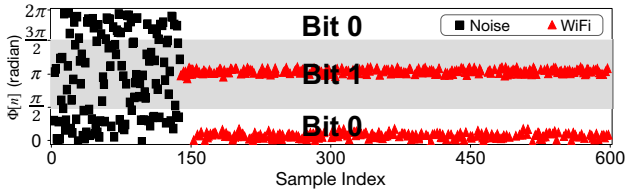


Figure 8: Narrowband decoding in practice.

Figure 8 demonstrates narrowband decoding result in practice, with decision boundaries in gray lines. It demonstrates that Φ is kept around π or 0 during WiFi packet, while it is random under noise, successfully validating the mechanism in practice. Narrowband decoding has multiple favorable practical features: as Eq. 2 suggests, it is as simple as tracking phase change, and it does not require synchronization as it holds at any point in time as long as the samples are $1\mu s$ apart. Furthermore, narrowband decoding universally applies to any WiFi and ZigBee channels combinations while robust to carrier frequency offset (CFO), which are analytically shown in Appendix. To sum up, aforementioned features significantly reduce the complexity, to keep narrowband decoding computationally light and viable for low-end ZigBee devices.

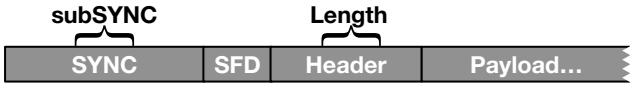


Figure 9: WiFi 802.11b packet structure. WiFi detection and guard band identification exploits SYNC and Header, respectively, located within the first 176bits.

4.3 Ultra-Lightweight WiFi Detection

G-Bee effectively and reliably detects incoming WiFi packets in real time, with only a single-bit comparison operation per sample (1Msps). Figure 9 illustrates WiFi packet structure beginning with the SYNC field containing fixed 128bits. As in the figure, G-Bee exploits this predefined bit sequence captured by narrowband decoding, where no bit error is tolerated. That is, G-Bee only captures error-free (i.e., high SNR) WiFi signals to utilize their guard bands. The rationale behind this is to maximize the chance of safeguarded communication – low SNR (i.e., low power) WiFi is less likely to make other WiFi devices to back off, and thus is less likely to safeguard ZigBee. Under this condition, a straight forward method to achieve SYNC detection is to store the last 128bits which are compared bit by bit with SYNC where WiFi is detected when all 128bits match. However, we note that this requires 128bit comparison for each incoming bit, or in other words, every $1\mu s$ (i.e., narrowband decoding rate), incurs significant overhead for low-end ZigBee devices.

We resolve this issue by only using the subsequence of SYNC for WiFi detection, which we refer to as *sub-SYNC*. This simple technique effectively minimizes computational and storage overhead to a *single bit* computation per μs (i.e., $\frac{1}{128}$ of 128bit computation), while meeting the reliability requirement. For brevity, this can be achieved simply by setting sub-SYNC to be a 38bit subsequence starting from 8th bit in SYNC. In the following we show how this can be derived.

Let us denote the SYNC as a bit vector $[b_0, b_1, ..., b_{127}]$. Without loss of generality we let sub-SYNC be a L-bit subvector, beginning from b_S: $[b_S, b_{S+1}, ..., b_{S+L-1}]$ where $0 \leq S \leq 127 - L$. Then the problem becomes finding S (start position) and L (length) that minimize the computational and storage overhead while meeting the reliability requirement.

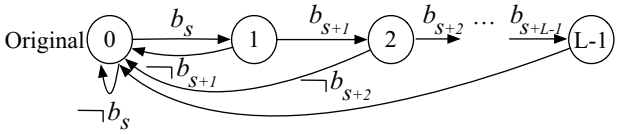


Figure 10: Finite state machine for sub-SYNC detection via single bit operation. Upon receiving a bit, it is compared to a sub-SYNC bit corresponding to the current state (i.e., 1 bit operation). Transition to the next state occurs if they match, or returns to state 0 otherwise. WiFi packet is detected upon reaching state $L-1$.

sub-SYNC Start Position. The mechanism of detection via single bit computation can be represented as a finite state machine (FSM) shown in Figure 10. For each decoded bit, the only operation is to compare that bit to the sub-SYNC bit corresponding to the current state. Transition to the next state occurs if they match, or returns to state 0 otherwise. WiFi packet is successfully detected upon reaching state $L-1$. We note that such an FSM is not able to detect sub-SYNC, if it is in a state other than 0 upon receiving sub-SYNC. For example, let's consider the received bit just before sub-SYNC accidentally matches with the first sub-SYNC bit (i.e., b_S), which could easily occur, with 50% probability. In this case, the FSM is in state 1 when sub-SYNC arrives, where bit mismatch will occur within sub-SYNC, resulting in a detection miss.

In order for the FSM to guarantee detection, it must be in state 0 with the beginning of sub-SYNC. Interestingly, this can be achieved simply by wisely selecting the start position, S. Conceptually, bits b_0 to b_{S-1} enforces mismatch at b_{S-1}, thus ensuring FSM to be in state 0 when sub-SYNC begins. For a rigorous analysis, we first discuss the two non-overlapped cases covering all detection miss scenarios. **Case #1:** Bit match occurs between b_0 and b_{S-1} and continues to b_{S-1}. **Case #2:** Bit match occurs before b_0 and continues to b_{S-1}. The following propositions provide features that sub-SYNC needs to have in order to avoid each cases.

Proposition 4.1. *Case #1 is avoided if* $[b_{S-1-n}, .., b_{S-1}] \neq [b_S, .., b_{S+n}]$ *for* $0 \leq n \leq S-1$.

Proof. The condition indicates a mismatch will always occur within $[b_0, ..., b_{S-1}]$. Let b_i be the mismatched bit, where $[b_{i+1}, ..., b_{S-1}]$ will also have mismatch. This repeats until $b_{S-1} \neq b_S$, therefore always keeping FSM at state 0 at the beginning of sub-SYNC. Therefore, Case #1 is avoided and proposition 4.1 holds. □

Proposition 4.2. *Case #2 is avoided if* $[b_0, ..., b_{S-1}]$ *is not a subsequence of sub-SYNC.*

Proof. By definition, Case #2 strictly requires the entire $[b_0, ..., b_{S-1}]$ to be matched in the FSM. This only occurs if it is a subsequence of sub-SYNC. Therefore, Proposition 4.2 holds. □

From Propositions 4.1 and 4.2, Cases #1 and #2 are avoided altogether if S is selected such that both the conditions in propositions are met; i.e., the FSM is guaranteed to capture sub-SYNC with only a single-bit computation per sample. Since the SYNC has fixed 128bits, feasible values of S are static and thus can be searched offline. Among multiple solutions, we select $S = 7$, the minimum value, for the earliest sub-SYNC detection and the longest skip time until reaching WiFi header (to be read by G-Bee), during which the node may be put to sleep for energy savings.

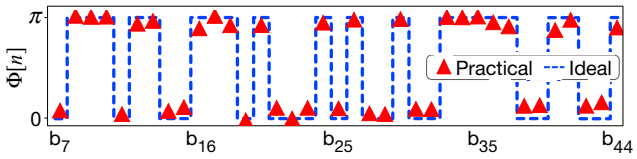


Figure 11: SubSYNC detection.

sub-SYNC Length. Reliability is determined by length L; false positive (FP: noise interpreted as WiFi) rate simply becomes 2^{-L} assuming random noise, while false negative (FN: detection miss) becomes $1 - (1 - BER)^L$. We set $L = 38$ for FP rate to be 3.6×10^{-12}, which translates to an average of a single error in more than 76 hours under $1\mu s$ narrowband sampling rate. Therefore, erroneous transmission vulnerable to interference is highly unlikely. On the other hand FN rate becomes 4.3% as measured under a moderate distance of 5m from WiFi transmitter, where it increases with weaker WiFi signal. This is acceptable as it naturally leads to infrequent use of the weaker WiFi signal, which is less likely to safeguard ZigBee.

To summarize, simply setting sub-SYNC of $[b_7, ..., b_{44}]$ enables WiFi detection under minimized computation and storage via 1bit computation, while meeting strict reliability constraints. Figure 11 illustrates an example of WiFi detection in practice.

4.4 Guard Band Identification

This sections discusses identifying the channel and duration of the guard band, or equivalently, finding the length and the channel of the WiFi detected via sub-SYNC matching. Recall that WiFi detection is initiated on the leftmost ZigBee channel (Figure 4a) overlapping with the WiFi channel (E.g., Ch_{12}^z for Ch_2^w). Upon WiFi detection, G-Bee switches to the rightmost channel (e.g., Ch_{15}^z for Ch_2^w) and resumes narrowband decoding from the start of the `Header` of the WiFi packet which begins at 145th bit. Essentially, `Header` interpretation provides complete information on the guard band, including duration and channel. As `Header` bits are scrambled as per the WiFi specification, narrowband decoded bits need be descrambled. If we let b_k is the k-th decoded bit from the start of the header, the corresponding descrambled bit, b_k^d is computed as:

$$b_k^d = (b_{k-7} \oplus b_{k-4}) \oplus b_k \quad (3)$$

Guard Band Duration. the last 16 descrambled bits (i.e., b_{160}^d-b_{175}^d) indicates the airtime of the WiFi packet represented in *us*, or in other words, the duration of the guard band. We note that the descrambling process is light both in computation (two XOR operations per bit) and storage (8bits stored), which we verify via empirical measurements in Section 7.

Guard Band Frequency. Guard band identification is performed based on the information obtained (i.e., the start of `Header`) from the WiFi detection. Therefore, successful `Header` interpretation strongly indicates that the corresponding WiFi packet overlaps with both leftmost and rightmost overlapping channels. This immediately identifies the WiFi channel the packet is on, as well the corresponding guard band. G-Bee effectively verifies interpretation success by simply matching known bits in the `Header` to the interpreted bits. This is feasible because the `Header` bits are scrambled – indicating that an error in one part propagates to the other. Careful observation of the `Header` contents and specification [9] yields 7 of such fixed bits, where five of them are reserved ($b_{152} - b_{153}$ and $b_{156} - b_{158}$) as bit '0'. The other two (b_{174}, b_{175}) are the last two bits of `Length` field. While they are not reserved, the maximum transmission unit (MTU) of 1,500 bytes only uses 14 bits out of the 16bit `Length` field, leaving the last two as bit '0'. Validation via the seven bits yields a high error detection rate of 99.2%.

Such validation also avoids G-Bee malfunction – Even a single bit error could lead to either not fully utilizing the guard band (if erroneously interpreted smaller), or transmitting for longer than the guard band duration when incorrectly interpreted to be longer, which puts the transmission under the potential risk of being interfered since it is no longer guarded for the full duration of the packet. In summary, validation error indicates either (i) WiFi packet under different channel or (ii) the signal is corrupted, where in both cases G-Bee aborts the transmission.

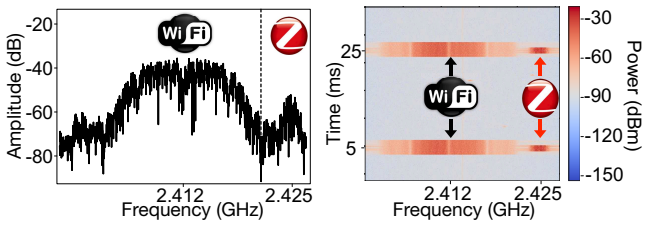


Figure 12: G-Bee transmission spectrum.

4.5 Safeguarded Communication

With the guard band channel and duration detected, G-Bee immediately switches to the transmission channel and initiates the transmission. The spectrum of G-Bee transmission is shown in Figure 12. Random delays may be applied for multiple G-Bee, which we discuss in later section. The maximum ZigBee packet size is dynamically computed depending on the guardband duration. That is, the size is $\frac{Length - T_t}{32}$ bytes, which ensures the ZigBee transmission is completed within the guardband duration, ensuring safety. We note that *Length* is the guardband duration (in μs) obtained earlier from the WiFi packet `Length` field and T_t is the ZigBee turn around time from Rx to Tx mode, which is $198\mu s$ in our device. Lastly, the denominator of $32\mu s$ is the per byte transmission delay of ZigBee.

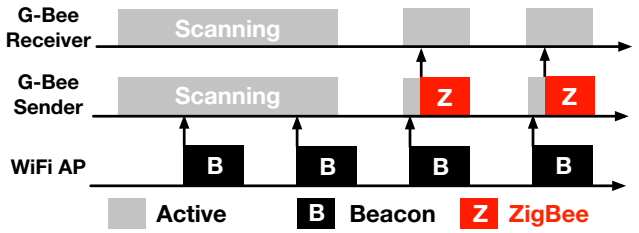


Figure 13: G-Bee scans and profiles the WiFi beacon period for a short time (=204.8*ms*) to utilize the beacon for the duty cycled transmission.

5 ENHANCED FEATURES

This sections discusses advanced features for low-duty cycle and multiple G-Bee support.

5.1 Beacon-synchronized Low Duty Cycle

Low duty cycle in G-Bee is achieved by leveraging (802.11b) beacons, which provides two advantages: (i) beacons are mandatory, enabling G-Bee in most of the real-life scenarios. (ii) Also, beacons are periodic. By default it repeats every 102.4ms, which is kept by the vast majority of the WiFi APs in practice [36]. This makes the timings of beacon transmissions to be easily estimated – or equivalently, the future occurrences of guard band (from beacons) becomes highly predictable. Leveraging on this predictability, G-Bee aligns the wakeup schedule with the predicted guard band timings (and sleeps in between), essentially *synchronizing* active time to the guard band (i.e., channel availability) and ensuring reliable transmission during the short active time. This effectively reduces the energy dissipation by avoiding unnecessary wakeup (when the channel is busy) and inherently mitigates the long delay due to packet loss, which is known to be one of the most critical downside in conventional low duty cycling. Figure 13 illustrates the low duty cycle operation in G-Bee, consisting of two steps: (1) beacon scanning and (2) low duty cycle transmission. Details are as follows.

Beacon Scanning. G-Bee utilizes two features to scan and profile WiFi beacon: (i) Unique length. Beacons embed information specific to WiFi APs (e.g. SSID), which result in different beacon lengths. In other words, the length is effectively a unique identifier of beacons from the same AP, which are kept on the same WiFi channel (and thus the channel need only be detected once). (ii) Second feature is the common beacon interval of 102.4*ms* [36].

G-Bee initiates beacon scanning by performing the basic (i.e., non-duty cycled) operation for 204.8*ms* (=102.4*ms* × 2), in which two beacons are captured from each AP. The arrival time, length, and channel of each 802.11b packet, which are naturally found by G-Bee operation, are recorded as three-element tuples. After 204.8*ms* the recorded tuples are sorted by packet lengths. If the two adjacent tuples have the same length and the arrival time difference of approximately 102.4*ms*, it is considered to be a beacon and is used for duty cycled transmission.

Low Duty Cycle Transmission. G-Bee node wakes up at the estimated arrival time of beacons (as scanned in the previous phase) and wait for the corresponding beacon; That is, if the length of the upcoming WiFi packet equals the length of expected beacon, G-Bee node switches to the guard band corresponding to this beacon (recorded in the previous phase) and start transmission for the beacon length duration, and returns to sleep immediately after transmission. When three WiFi beacons from the same AP is consecutively missing, mobility or environmental change is assumed and G-Bee reinitiates beacon scanning.

5.2 Supporting Multiple G-Bee

This section discusses how multiple G-Bees are supported in a fair and efficient manner. When multiple G-Bee nodes detect the same WiFi guard band on the fly, they are naturally synchronized and will result in collision if two or more of them transmit on the the guard band. To avoid this G-Bee introduce a random backoff within the size of the contention window, followed by CCA. A node with smallest backoff delay accesses the channel, where the contention window is dynamically adjusted based on distributed multiplexing approaches. Specifically, we introduce two mechanisms either focused on fairness or energy/spectrum efficiency.

Fairness Prioritized. The window size is halved whenever a node loses in the contention, increasing the chance of channel access. This offers unbiased, fair access to all nodes.

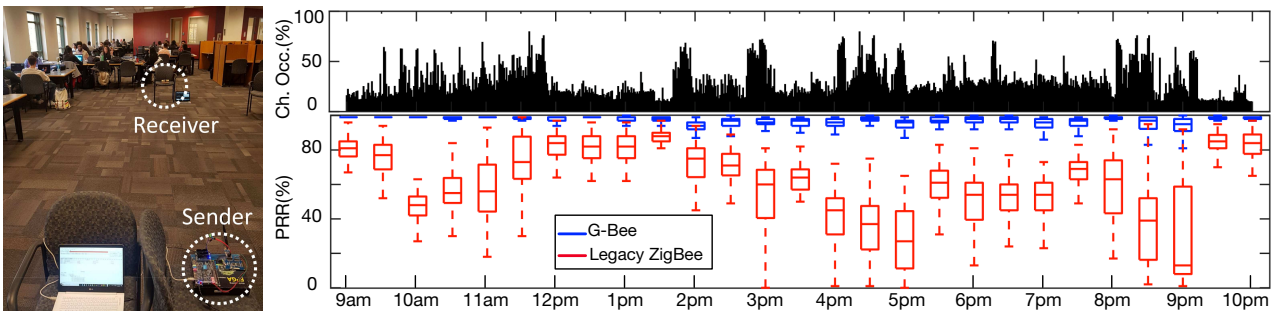
Energy/Spectrum Efficiency Prioritized. The G-Bee node with the data to be transmitted that most 'fits' within the guard band duration (i.e., which makes the most out of spectrum) has a higher priority. That is, contention window size is directly computed as the difference between the captured 802.11b packet size and the size of the ZigBee packet to be transmitted.

We also note that a WiFi channel has two guard bands, on the right and left side. Therefore, G-Bee is able to support two ZigBee packets per WiFi packet. To summarize, under n WiFi (802.11b) pkts/s, G-Bee (with either of our multiple access designs) safely supports $2n$ ZigBee pkts/s and the corresponding number of devices – That is, $2n$ devices each with 1 pkt/s, which is a reasonable estimation given the low ZigBee traffic volume in practice. According to our measurement of > 50 WiFi 802.11b pkts/s (Section 7.4), this translates to > 100 supported devices. We note that this is a fair number of devices especially given the limited transmission range of ZigBee.

6 DISCUSSION

This section presents practical considerations for G-Bee.

Coexistence with 802.11g/n. G-Bee fully maintains its functionality under all three WiFi variants on 2.4GHz: 802.11b, g and n. That is, despite the difference in the bandwidths of 802.11g (20MHz) and n (20/40MHz) from 802.11b (22MHz), due to the channel layout the guard band of the 802.11b is kept interference-free. For instance when 802.11b traffic is on Ch_1^w, the closest non-overlapped 802.11g (20MHz) and n (40MHz) channels are centered at Ch_6^w and Ch_8^w respectively. This provides 4MHz guardband, sufficiently wide to protect 2MHz ZigBee.



(a) Library setting

(b) Upper: Channel occupancy by time, Lower: G-Bee vs. legacy ZigBee PRR

Figure 14: PRR of G-Bee and legacy ZigBee deployed in a university library. G-Bee achieves reliable communication throughout the day, even under hundreds of WiFi users.

Compatibility with Legacy ZigBee. Benefiting from transparency to ZigBee receiver, backward compatibility to legacy ZigBee receiver is naturally achieved in G-Bee. Also, G-Bee sender senses channel before transmitting, indicating CSMA still remains intact. In short, G-Bee is fully compatible and non-disruptive to legacy ZigBee networks, ensuring low deployment cost without modification to readily deployed networks.

Hidden node terminal. When WiFi beacon are exploited, G-Bee nodes maintain beacon timing table (Section 5.1) with the different APs and the estimation of arrival time of the corresponding beacons. Hidden terminal problem occurs when the G-bee is in the range of both WiFi AP and interference source (e.g., another WiFi), while the interference source is outside the range of the WiFi AP. Then the G-Bee transmission is exposed to the interference and is no longer protected. To address this, G-Bee first detects the hidden terminal effect from consecutive packet corruptions. Then the corresponding AP is blacklisted from the AP beacon timing table to avoid further use of its beacons. We leave the implementation of hidden terminal prevention as our future work.

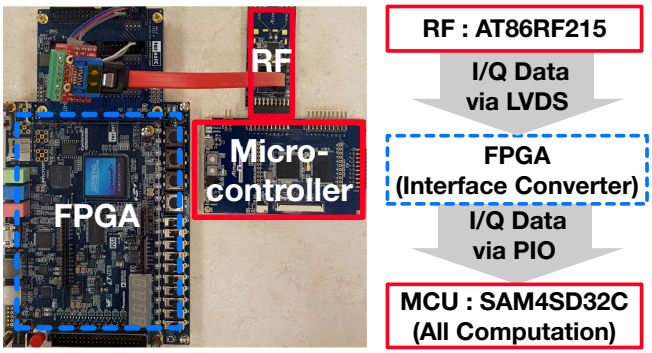


Figure 15: G-Bee platform built from off-the-shelf devices. FPGA only used as interface converter (LVDS → PIO), where all computation is performed on the microcontroller.

7 PERFORMANCE EVALUATION

This section presents G-Bee device prototype followed by extensive performance evaluations under various scenarios.

7.1 G-Bee Platform Prototype

As depicted in Figure 15, our prototype, supporting real-time operation, consists of three off-the-shelf components: (i) ZigBee-compliant RF front-end (AT86RF215) for communication, (ii) microcontroller (SAM4SD32C) performing all computation associated with G-Bee, and (iii) FPGA that only serves as a I/Q sample interface converter (LVDS → PIO) between RF front-end and microcontroller, due to their lack of interface compatibility. To ensure that our design runs entirely on software via the microcontroller, *no computation* is performed on the FPGA, which will be shown via measurement studies (Section 7.6). Furthermore, while we use the FPGA for flexibility, we note that its function can easily be integrated in the custom microcontroller board.

7.2 Safeguarding against Interference

Here we demonstrate G-Bee's reliability under saturated CTI.

Real-life Scenario (Library). We further evaluate G-Bee reliability in a practical setting of a university library. Figure 14a shows the deployment where we run the experiment for the entire day from 9AM to 10PM to reflect the dynamics of the number of students (i.e., WiFi users) and varying WiFi traffic and pattern throughout the day. Over 18APs are already installed in the library.

G-Bee sender and receiver pair are $5m$ apart with $0dBm$ TX power. We place a commercial WiFi AP (WNDR3800) in the library to increase the G-Bee packet transmission rate to 50pkts/s. The rate was intentionally set high (compared to practical ZigBee) to obtain a reliable result, largely avoiding the potential statistical bias due to dynamics of WiFi interference. Figure 14b depicts the result, where the upper figure shows the degree of WiFi traffic represented in channel occupancy, measured as the rate of channel energy above $-85dBm$ (typical WiFi signal strength). The lower figure shows that G-Bee reliably achieves an average PRR over 96.7% during the entire day, with low variance. On the contrary, the performance of legacy ZigBee is highly dependent on the WiFi traffic, fluctuating from 15% (9PM) to 88% (1:30PM) PRR. G-Bee consistently outperforms legacy ZigBee, reaching over 6.3× at 9PM.

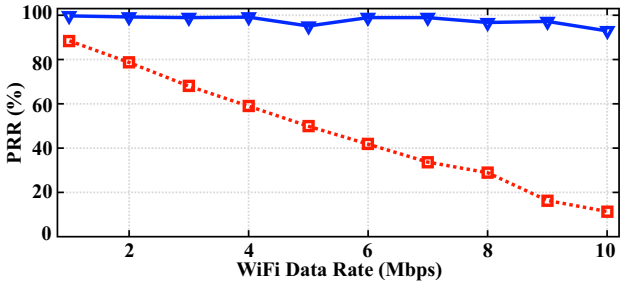


Figure 16: G-BEE consistently achieves over 93% PRR regardless of WiFi traffic, while that of the legacy ZigBee drops significantly to 14%.

Controlled Scenario (Lab). We demonstrates the reliability of G-Bee compared to legacy ZigBee under different degrees of WiFi traffic. The experiment is conducted in an empty lab at a university building. G-Bee and legacy ZigBee are both set to Ch_{12}^{z} at the transmission power of 0dBm, and are placed $10m$ away from WiFi AP disseminating traffic at Ch_{1}^{w}. `Iperf` is used to control the WiFi traffic load from 1-10Mbps throughput, where 1,000 G-Bee/ZigBee packets at 10Hz transmission rate are tested for each WiFi throughput setting. Figure 16 demonstrates that G-Bee consistently achieves 92.9-99.7% PRR, even under the heavy WiFi traffic of 10Mbps corresponding more than 800 WiFi packets per second. On the other hand PRR of the legacy ZigBee drops significantly to 13.6% at 10Mbps WiFi traffic, indicating that it is largely inoperable.

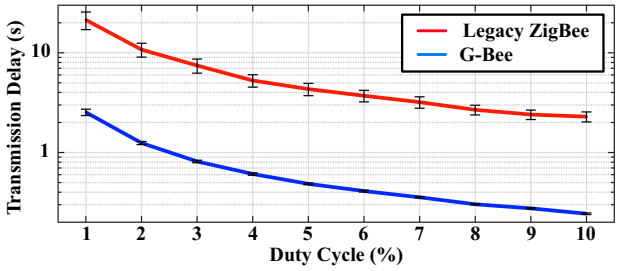


Figure 17: G-Bee reduces the low duty cycle delay to less than 87.5% of legacy ZigBee.

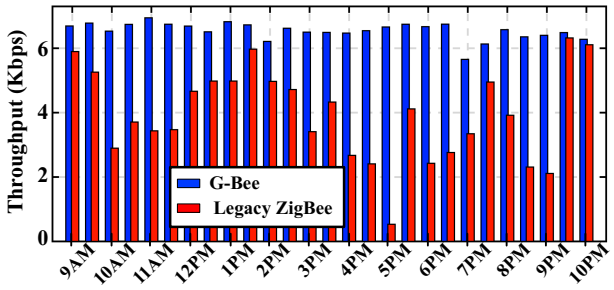


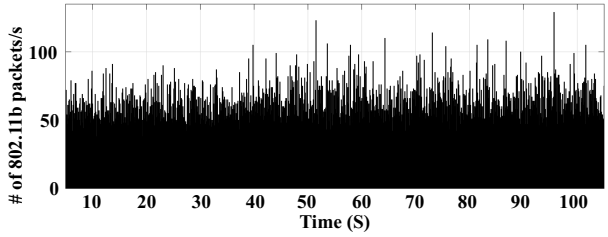
Figure 18: G-Bee achieves reliable throughput (average 6.5Kbps) with the low duty cycle.

7.3 Duty Cycle Efficiency

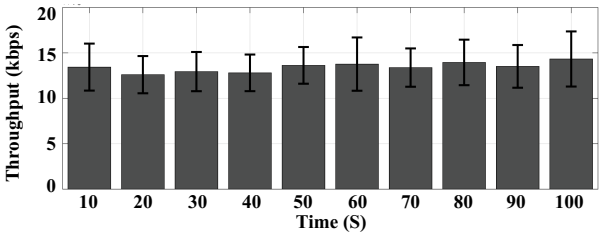
Delay. While duty cycling offers significant energy savings, large delay is known to be the main drawback for low duty cycling operation [12], which could be greatly aggravated under low communication reliability. G-Bee, by ensuring reliable communication, has the potential to resolve the critical issue of extensive delay. We run an experiment in the real-life setting of university library, where the duty cycle is adjusted between 1 and 10% with the active duration of $20ms$. The results are shown in Figure 17, where G-Bee suppresses the delay to less than $\frac{1}{8}$ of that of the legacy ZigBee at 1% of duty cycle. In sum, the result indicates that G-Bee, by ensuring reliable communication, effectively keeps the delay low and stable, essentially serving as a key enabler for low duty cycle operation under severe interference.

Throughput. We measure the low duty cycled throughput at university library for G-Bee and legacy ZigBee. To ensure a fair comparison, keep all parameters consistent between the two – 50pkts/s where each of packet are 21bytes in length and duty cycle of 2% with $5m$ distance. From Figure 18, G-Bee reaches an average 6.5Kbps throughput where legacy ZigBee shows average 3.9Kbps. We observe that the throughput variation of 1.3Kbps in G-Bee is mainly due to WiFi signal blockage by people, which lowers the WiFi signal power so as to be undetectable by G-Bee. However, we argue that this is not an issue – low SNR WiFi is unlikely to safeguard G-Bee transmission and therefore should not be used, in order to maintain the energy efficiency of the low duty cycle.

(a) Washington, D.C. Downtown experiment setting.



(b) 802.11b WiFi packets per second ($@Ch_{1}^{w}$).



(c) G-Bee throughput.

Figure 19: G-Bee throughput in downtown Washington, D.C.

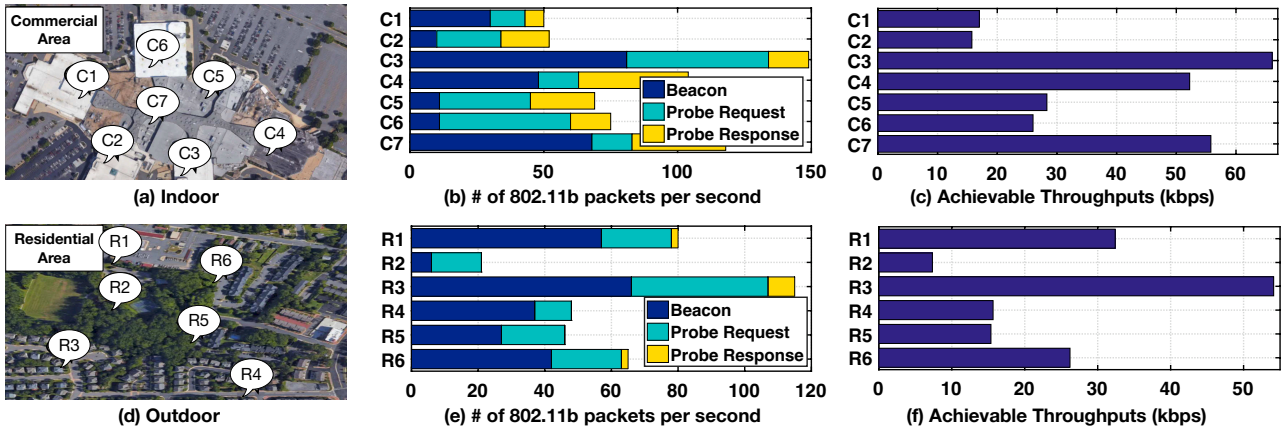


Figure 20: Indoor and outdoor opportunities.

7.4 Throughput via Ambient 802.11b

In this section we evaluate G-Bee's throughput under real-life network environment.

Throughput in downtown Washington, D.C. As shown in Figure 19a, the measurement is performed at a coffee shop in Washington, D.C. downtown, in order to demonstrate 802.11b availability and corresponding throughput under a practical scenario. G-Bee Tx transmits packets to Rx at a distance of 7m with 0dBm Tx power. At the same time, the ambient WiFi packets on the Ch_1^w are monitored by Wireshark. As shown in Figure 19b, Ch_1^w has an average of 64 802.11b pkts/s where 17APs using 802.11b control packets were found. More importantly, five of these APs are using 802.11b beacons which are periodically transmitted. Therefore, G-Bee, as demonstrated in Figure 19c, is able to consistently maintain a throughput of higher than 10.6Kbps due to the periodic opportunities.

Opportunities Under Diff. Scenarios. We collect 802.11b packets at different locations in 2 scenarios: commercial and residential area for trace-driven experiment. The 802.11b traffic and G-Bee throughput are shown in Figure 20. The number of 802.11b control packets is greater than 50pkts/s, providing more than 17.2Kbps for G-Bee communication even in the parking lot (C_1). The throughput of residential area are 32.4, 7.3, 54.1, 15.6, 15.3 and 26.2 for R_1~R_6 respectively. The results show that an average of 76 802.11b packets are presented in both shopping malls and residential areas that can represent real-world scenarios, and G-Bee achieves an average throughput of more than 31.7 Kbps, which is sufficient to support most IoT applications.

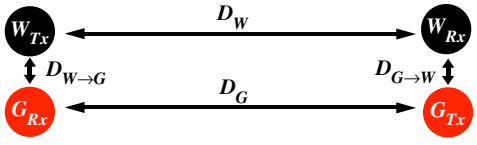
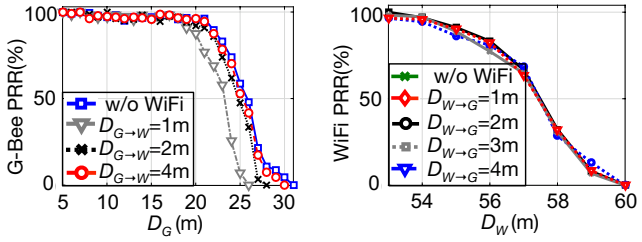


Figure 21: Cross-impact experiment setting of WiFi and G-Bee devices.



(a) G-Bee PRR, diff. $D_{W\to G}$ **(b) WiFi PRR, diff. $D_{G\to W}$**

Figure 22: Cross-impact between WiFi ($@Ch_{12}^w$) and G-bee ($@Ch_{26}^z$).

7.5 Impact of G-Bee & WiFi Ch. Adjacency

Here we study the cross-impact between WiFi and G-Bee. We use the PRR to show such influence caused by signal leakage from WiFi and Zigbee channels. We set up a WiFi link on Ch_{12}^w and a G-Bee link on Ch_{26}^z in a lobby at a university building. As shown in Figure 21, WiFi Tx(NETGEAR WNDR3800)-Rx(laptop) and G-Bee Tx-Rx pair are set to $[0, 60m]$ and $[0, 30m]$, respectively. $D_{G\to W}$ and $D_{W\to G}$ represent the distance between WiFi and G-Bee. While WiFi Tx transmits 1000 packets of 20dBm to WiFi Rx, G-Bee also sends 1000 packets to G-Bee Rx with 5dBm transmission power using the corresponding guard band.

Impact on G-Bee. Figure 22a shows the impact of adjacent WiFi on G-Bee. G-bee PRR drops significantly from 100% to 50% when D_G reaches over 20m due to low SNR. Especially, the PRR at $D_{W\to G} = 1m$ and $D_G = 25m$ becomes 7.1%, 10× smaller than without WiFi, showing that G-Bee Rx nodes are affected by WiFi Tx installed at a distance of less than $1m$. On the other hand, the WiFi can affect G-Bee's communication only if the $D_{W\to G} \leq 1m$ which rarely happens in practice.

Impact on WiFi. Figure 22b depicts the influence of G-Bee on WiFi, which is found to be negligible even when the G-Bee Tx is placed very close (=$1m$) to WiFi Rx. In other words, the impact of signal leakage of ZigBee to WiFi is minimal, indicating that G-Bee is non-disruptive to existing WiFi.

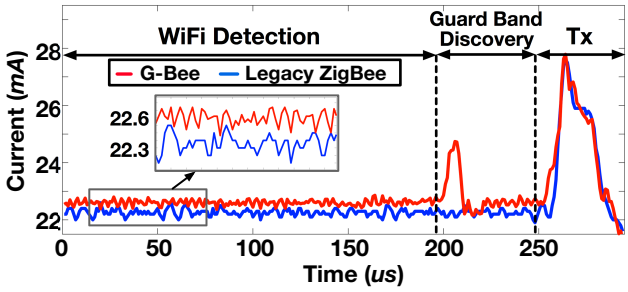


Figure 23: G-Bee power consumption comparison with legacy ZigBee.

7.6 Energy and Computational Cost

This section demonstrates power consumption and computational cost of G-Bee.

Power Consumption. Recall that G-Bee consists of three steps: (i) light-weight WiFi detection, (ii) guard band identification, and (iii) transmission. We measure the power consumption of the three steps directly from the G-Bee prototype. We also put the device in the legacy ZigBee mode and measure the power consumption of idle listening (corresponding to steps (i) and (ii) in G-Bee) and transmission. This is done using Tektronix MDO 3024 oscilloscope via microcontroller current measurement pin [28].

G-Bee, through steps (i) and (ii), incurs additional computation compared to legacy ZigBee. Step (i) requires one bit decoding and comparison according the FSM every 1us. Figure 23 shows the measured current from the our prototype (RF and microprocessor). The average current of a legacy ZigBee (idle listening) and G-Bee (step (i)) are 22.3mA and 22.6mA, respectively, indicating only 0.3mA or 1.3% increase. Descrambling and channel switching in step (ii) consumes an average of 22.9mA, 2.6% higher compared to legacy ZigBee. Step (iii) is identical for G-Bee and legacy ZigBee. We also consider additional 0.6mA consumed by the FPGA for interface conversion. Let us compare the energy dissipation between G-Bee and legacy ZigBee when transmitting a 30 byte packet. G-Bee slightly increases the energy consumed per packet compared to legacy (75.6mW vs. 73.6mW) while bringing a large difference in reliability, which results in a significant gain in overall energy consumption. Specifically, legacy ZigBee consumes 490.6mW (=73.6mW×6.6) for a packet delivery, where an average of 6.6 transmissions are needed until a success (according to our measurement under heavy traffic environment). Meanwhile, G-Bee uses 83.2mW (=75.6mW×1.1) with only 1.1 expected transmission per packet. This shows that the G-Bee saves 83% of energy compared to legacy ZigBee.

Computational Overhead. G-Bee's light computation can be easily performed on a moderate low-power device. Do observe this, we precisely measure WiFi detection and guard band identification execution time by triggering GPIO (General Purpose Input/Output) pin, captured by the oscilloscope. In our prototype equipped with a microprocessor running at 128MHz, G-Bee imposes 13cycles (101.5ns) of computation per 1us, indicating 10.2%. This translates to a negligible overhead of 0.3% under a practical scenario, since (i) ZigBee's traffic load is generally light (3pkts/s in our computation) indicating that the computation occurs infrequently and (ii) only an average of 10ms is needed to detect the guard band under 50 802.11b pkts/s, following our measurement. In other words, G-Bee operation occurs for only 10ms per packet transmission. This light overhead enables G-Bee to run on low end IoT devices without disrupting the system.

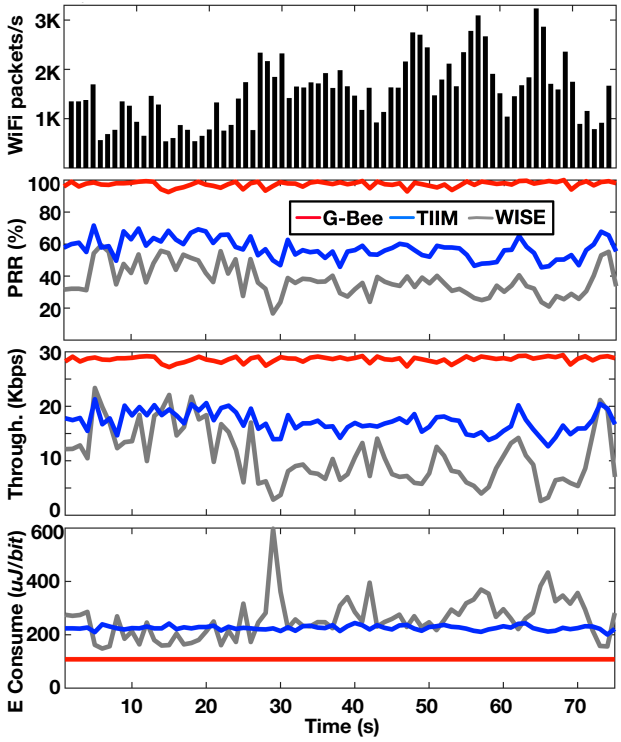
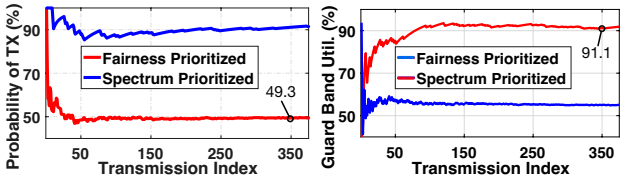


Figure 24: G-Bee and other techniques comparison (PRR, throughput, and energy consumption).

7.7 Performance Comparison

We also compare G-Bee to two known techniques in literature, WISE [17] and TIIM [16], in terms of PRR, throughput, and energy consumption via trace-based simulation. While G-Bee uses the deterministic opportunities of the guard band, both WISE and TIIM analyze the interference pattern to improve the performance of the ZigBee. The top figure in 24 shows the saturated WiFi traffic collected in an office for 75 seconds. WiFi channels are occupied with significant traffic (average of 2000 pkts/s), resulting in low PRR or $\leq$ 60% and $\leq$ 73% for WISE and TIIM, respectively. Similarly, WISE showed throughput of $\leq$ 25Kbps where TIIM reached $\leq$ 22Kbps. More importantly, the performance of the two techniques fluctuate with the pattern and degree of WiFi interference, indicating low reliability. Conversely, G-Bee demonstrates reliable and high PRR of 97.3% and throughput of 28.8Kbps regardless of WiFi traffic. In terms of energy efficiency, G-Bee consumes an average of 108uJ/bit, which is 58.1% and 52.0% reduction from WISE (258uJ/bit) and TiMM (225uJ/bit)], respectively. To sum up, G-Bee not only outperforms the state-of-the-art techniques, but also is highly reliable under saturated WiFi traffic.



(a) Fairness. (b) Spectrum.

Figure 25: G-Bee achieves fair (49.3%) and spectrum efficient (91.1%) transmission between multiple G-Bee nodes.

7.8 Multiple G-Bee

Two channel access approaches (fairness and energy/spectrum efficiency prioritized) are evaluated under the simple scenario of two G-Bee nodes. The results are depicted in Figures 25a and 25b, where packet length of two G-Bee nodes are 38Bytes and 6Bytes at a distance of 3m. After they detect the 802.11b packet simultaneously, 2 nodes competes for accessing the channel. The node which has a shorter random delay transmits a packet. Two nodes with the fairness technique can achieve 49.3% of transmission opportunities, indicating that each node has the same probability to access the channel. Figure 25a effectively demonstrates that fairness prioritized mechanism offers even transmission opportunities to both nodes. On the other hands, two nodes with the spectrum efficiency technique can utilize the guard band 91.1% due to higher priority of a longer packet. Therefore, spectrum prioritized mechanism achieves higher utilization of the guard band, indicating better spectrum efficiency.

8 RELATED WORK

There had been many efforts in the wireless community to resolve interference. Those works focus on the following 4 directions: (i) interference cancellation [5, 7, 8, 13, 24, 39, 41] captures (cross-technology) interference and remove it by using high power and high sampling-rate SDR. (ii) Corruption recovery retransmits the corrupted part whenever there exists interference [15] or makes ZigBee packets more robust via coding [22, 26]. However, corruption recovery sacrifices spectrum and energy. (iii) Instead of dealing with interference passively, white space networking [18, 19, 21, 38, 43], estimates channel usage via traffic modeling or generating white space through signaling and multi-channel cooperation. (iv) Carefully coordinated services such as NB-IoT in the licensed band provide interference avoidance by resource (i.e., spectrum) allocation rather than estimating channel traffic [3]. Although NB-IoT utilizes the guard band precisely, it requires to share the same hardware and technology to coordinate with LTE. This is not applicable to the unlicensed ISM band where heterogeneous technologies access the spectrum distributively. Different from all previous work, G-Bee explores spectrum availability in a completely passive and deterministic manner, without affecting existing networks. To achieve this G-Bee leverages physical-layer information (I/Q) accessible in commercial ZigBee radio chip, while keeping the overhead minimal for practicality to low-end IoT devices.

9 CONCLUSION

This paper presents G-Bee that safely guards ZigBee delivery even under saturated WiFi traffic. By using physical layer information (I/Q) in commodity ZigBee RF chips, G-Bee detects ambient WiFi packet in an extreme light-weight way, recognizes its duration& channel, and then transmits ZigBee packets on the guard band of ongoing WiFi traffic, which turns ambient WiFi interference into protection. Further more, G-Bee synchronizes low duty cycling with guard band availability for maximized energy efficiency. Extensive testbed experiments demonstrate that G-Bee consistently achieves $\geq$ 95% PRR, more than 6.3× compared to legacy ZigBee. Moreover, G-Bee outperforms the state-of-the-art technique by 2.5× in terms of PRR, throughput, and energy.

APPENDIX: IMPACT OF FREQ. OFFSET

Universality to ZigBee channels. There are 16 WiFi channels, where each overlaps with four ZigBee channels. We show that the narrowband decoding is universally applied regardless of the WiFi and ZigBee channel combinations. In other words, Φ is kept at $\{\pi,0\}$ regardless of WiFi channel (f_w) and ZigBee channel (f_z). Plugging $4(n-1)$ into equation 1 we get $s[4(n-1)]$:

$$\begin{aligned} s[4(n-1)] &= \left(w(4(n-1)T_s)e^{j2\pi(f_w-f_z)4(n-1)T_s}\right) * h \\ &= \left(w(4(n-1)T_s)e^{j2\pi(f_w-f_z)4nT_s}\underline{e^{j8\pi(f_z-f_w)T_s}}\right) * h \end{aligned} \quad (4)$$

Note that $T_s = 250ns$ and the channel frequency difference between WiFi and ZigBee (i.e., $f_z - f_w$) is either -7, -2, 3, or 8 MHz according to the WiFi and ZigBee channels layout defined in 802.11 [9] and 802.15.4 [10] standards. This leads to the underlined term become 1, regardless of channel frequency difference. Then, we obtain:

$$s[4(n-1)] = \left(w(4(n-1)T_s)e^{j2\pi(f_w-f_z)4nT_s}\right) * h \quad (5)$$

Hence, $s[4(n-1)] = s[4n]$ when $w(4(n-1)T_s) = w(4nT_s)$. Otherwise, $s[4(n-1)] = -s[4n]$. Therefore, the phase difference between $s[4(n-1)]$ and $s[4n]$, i.e. $\Phi[n]$, yields 0 when $w(4(n-1)T_s) = w(4nT_s)$ (i.e., bit = 0) and π when $w(4(n-1)T_s) = -w(4nT_s)$ (i.e., bit = 1), regardless of channel frequency difference.

Robustness to CFO. Φ can potentially be affected by carrier frequency offset (CFO), where a recent study reveals an average CFO (for commercial ZigBee radio) of ±2KHz [11]. This yields $\pm\frac{\pi}{250}$ error, only 0.8% of the error tolerance ($=\frac{\pi}{2}$) for Φ, indicating that the effect of CFO on G-Bee is minimal.

ACKNOWLEDGMENTS

This work was supported in part by the NSF under grant CNS-1717059. We sincerely thank the shepherd and reviewers for their valuable feedback and suggestions.

REFERENCES

[1] Atmel Corporation. 2016. AT86RF215. http://www.atmel.com/Images/Atmel-42415-WIRELESS-AT86RF215_Datasheet.pdf.

[2] Dave Cavalcanti, Sushanta Das, Jianfeng Wang, and Kiran Challapali. 2008. Cognitive radio based wireless sensor networks. In *Computer Communications and Networks, 2008. ICCCN'08. Proceedings of 17th International Conference on*. IEEE, 1–6.

[3] Min Chen, Yiming Miao, Yixue Hao, and Kai Hwang. 2017. Narrow band internet of things. *IEEE Access* 5 (2017), 20557–20577.

[4] Daniele Croce, Natale Galioto, Domenico Garlisi, Fabrizio Giuliano, and Ilenia Tinnirello. 2017. An inter-technology communication scheme for wifi/zigbee coexisting networks. In *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing, 305–310.

[5] Behnam Dezfouli, Marjan Radi, and Octav Chipara. 2016. Real-time communication in low-power mobile wireless networks. In *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*. IEEE, 680–686.

[6] Dialog Semiconductor. 2014. DA14681. https://support.dialog-semiconductor.com/downloads/DA14580_DS_v3.1.pdf.

[7] Shyamnath Gollakota, Fadel Adib, Dina Katabi, and Srinivasan Seshan. 2011. Clearing the RF Smog: Making 802.11N Robust to Cross-technology Interference. In *Proceedings of the ACM SIGCOMM 2011 Conference (SIGCOMM '11)*.

[8] Shyamnath Gollakota and Dina Katabi. 2008. Zigzag decoding: combating hidden terminals in wireless networks. In *Proceedings of the ACM SIGCOMM 2008 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Seattle, WA, USA, August 17-22, 2008*. 159–170.

[9] IEEE 802.11 Working Group et al. 2010. IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 6: Wireless Access in Vehicular Environments. *IEEE Std* 802, 11 (2010).

[10] IEEE 802.11 Working Group et al. 2011. IEEE Std 802.15.4-2011, IEEE Standard for Local and metropolitan area networks. (September 2011).

[11] Fujuan Guo, Baofeng Zhou, and Mehmet C Vuran. 2017. CFOSynt: Carrier frequency offset assisted clock syntonization for wireless sensor networks. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 1–9.

[12] Shuo Guo, Liang He, Yu Gu, Bo Jiang, and Tian He. 2014. Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. *IEEE Trans. Comput.* 63, 11 (2014), 2787–2802.

[13] Daniel Halperin, Thomas E. Anderson, and David Wetherall. 2008. Taking the sting out of carrier sense: interference cancellation for wireless LANs. In *Proceedings of the 14th Annual International Conference on Mobile Computing and Networking, MOBICOM 2008, San Francisco, California, USA, September 14-19, 2008*. 339–350.

[14] Haitham Hassanieh, Lixin Shi, Omid Abari, Ezzeldin Hamed, and Dina Katabi. 2014. Ghz-wide sensing and decoding using the sparse fourier transform. In *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2256–2264.

[15] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy. Crosszig. 2016. Combating cross-technology interference in low-power wireless networks. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE, 1–12.

[16] Anwar Hithnawi, Hossein Shafagh, and Simon Duquennoy. 2015. TIIM: technology-independent interference mitigation for low-power wireless networks. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM, 1–12.

[17] Jun Huang, Guoliang Xing, Gang Zhou, and Ruogu Zhou. 2010. Beyond coexistence: Exploiting WiFi white space for Zigbee performance assurance. In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*. IEEE, 305–314.

[18] Jun Huang, Guoliang Xing, Gang Zhou, and Ruogu Zhou. 2010. Beyond coexistence: Exploiting WiFi white space for ZigBee performance assurance. In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*. 305–314.

[19] Min Li Huang and Sin-Chong Park. 2009. A WLAN and ZigBee Coexistence Mechanism for Wearable Health Monitoring System. In *Proceedings of the 9th international conference on Communications and information technologies (ISCIT)*.

[20] Texas Instruments. 2009. CC253x System-on-Chip Solution for 2.4-GHz IEEE 802.15. 4 and ZigBee Applications. *Literature Number: SWRU191* (2009), 54–56.

[21] Byoung Hoon Jung, Jo Woon Chong, Chang Yong Jung, Su Min Kim, and Dan Keun Sung. 2008. Interference Mediation for Coexistence of WLAN and ZigBee Networks. In *Personal, Indoor and Mobile Radio Communications (PIMRC)*.

[22] Sachin Katti, Shyamnath Gollakota, and Dina Katabi. 2007. Embracing wireless interference: Analog network coding. *ACM SIGCOMM Computer Communication Review* 37, 4 (2007), 397–408.

[23] Nacer Khalil, Mohamed Riduan Abid, Driss Benhaddou, and Michael Gerndt. 2014. Wireless sensors networks for Internet of Things. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*. IEEE, 1–6.

[24] Linghe Kong and Xue Liu. 2015. mZig: Enabling Multi-Packet Reception in ZigBee. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, MobiCom 2015, Paris, France, September 7-11, 2015*. 552–565.

[25] Nicholas D Lane, Sourav Bhattacharya, Akhil Mathur, Petko Georgiev, Claudio Forlivesi, and Fahim Kawsar. 2017. Squeezing Deep Learning into Mobile and Embedded Devices. *IEEE Pervasive Computing* 16, 3 (2017), 82–88.

[26] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. 2010. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 309–322.

[27] Feng Lu, Geoffrey M Voelker, and Alex C Snoeren. 2013. SloMo: Downclocking WiFi Communication.. In *NSDI*. 255–268.

[28] SMART ARM-based Microcontrollers. [n. d.]. SAM4S Xplained Pro. ([n. d.]).

[29] Mobashir Mohammad, XiangFa Guo, and Mun Choon Chan. 2016. Oppcast: Exploiting spatial and channel diversity for robust data collection in urban environments. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*. IEEE Press, 19.

[30] Semtech. 2018. SX1257. https://www.semtech.com/uploads/documents/DS_SX1257_V1.2.pdf.

[31] Vijayendra Kumar Sharma and Dheerendra Singh. 2014. QOS Based Power Management in ZIGBEE Wireless Sensor Network. (2014).

[32] Lixin Shi, Paramvir Bahl, and Dina Katabi. 2015. Beyond Sensing: Multi-GHz Realtime Spectrum Analytics.. In *NSDI*. 159–172.

[33] Texas Instruments. 2013. CC2420. http://www.ti.com/lit/gpn/cc2420.

[34] Texas Instruments. 2014. CC1200. http://www.ti.com/lit/ds/symlink/cc1200.pdf.

[35] Texas Instruments. 2018. CC1310. http://www.ti.com/lit/ds/symlink/cc1310.pdf.

[36] Sudarshan Vasudevan, Konstantina Papagiannaki, Christophe Diot, Jim Kurose, and Don Towsley. 2005. Facilitating access point selection in IEEE 802.11 wireless networks. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. Usenix Association, 26–26.

[37] Dong Wang, Boleslaw K Szymanski, Tarek Abdelzaher, Heng Ji, and Lance Kaplan. 2018. The age of social sensing. *arXiv preprint arXiv:1801.09116* (2018).

[38] Yufei Wang, Qixin Wang, Zheng Zeng, Guanbo Zheng, and Rong Zheng. 2011. WiCop: Engineering WiFi Temporal White-Spaces for Safe Operations of Wireless Body Area Networks in Medical Applications. In *Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd*.

[39] Yubo Yan, Panlong Yang, Xiang-Yang Li, Yafei Zhang, Jianjiang Lu, Lizhao You, Jiliang Wang, Jinsong Han, and Yan Xiong. 2015. WizBee: Wise ZigBee Coexistence via Interference Cancellation with Single Antenna. *IEEE Trans. Mob. Comput.* 14, 12 (2015), 2590–2603.

[40] Rong Yu, Yan Zhang, Stein Gjessing, Chau Yuen, Shengli Xie, and Mohsen Guizani. 2011. Cognitive radio based hierarchical communications infrastructure for smart grid. *IEEE network* 25, 5 (2011).

[41] Yan Yubo, Yang Panlong, Li Xiangyang, Tao Yue, Zhang Lan, and You Lizhao. 2013. ZIMO: Building Cross-technology MIMO to Harmonize ZigBee Smog with WiFi Flash Without Intervention. In *Proceedings of the 19th Annual International Conference on Mobile Computing (MobiCom '13)*.

[42] Peilin Zhang, Olaf Landsiedel, and Oliver Theel. 2017. MOR: Multichannel Opportunistic Routing for wireless sensor networks. In *Proc. of EWSN*.

[43] Xinyu Zhang and Kang G. Shin. 2011. Enabling Coexistence of Heterogeneous Wireless Systems: Case for ZigBee and WiFi. In *MobiHoc*.

[44] Xia Zhou, Zengbin Zhang, Gang Wang, Xiaoxiao Yu, Ben Y Zhao, and Haitao Zheng. 2013. Practical conflict graphs for dynamic spectrum distribution. In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 41. ACM, 5–16.