Symbol-level Cross-technology Communication via Payload Encoding

Shuai Wang[*], Song Min Kim [*‡], and Tian He[†‡]

[*]Department of Computer Science, George Mason University, USA

[†]Department of Computer Science and Engineering, University of Minnesota, USA

[‡]Co-corresponding authors

{swang42, song}@gmu.edu, tianhe@cs.umn.edu.

Abstract—To mitigate the issue of cross-technology interference (CTI) under dense wireless, cross-technology communication (CTC) was recently proposed, which enables direct communication among heterogeneous wireless technologies. We present SymBee, a novel ZigBee to WiFi CTC with symbol-level encoding for performance breakthrough from state-of-the-arts. SymBee is uniquely built on the new insight on ZigBee-WiFi physical layer interaction *cross-observability* – i.e., the output on WiFi when fed with ZigBee signal (due to frequency overlap). This is analyzed experimentally and theoretically through rigorous derivations, from which the key innovation in SymBee design, i.e., payload encoding, stems; Conveying data across technologies is as simple as putting specific symbols in ZigBee packet payload, such that they yield unique and easily detectable patterns when cross-observed at WiFi. This symbol-level encoding is fully compatible with any commodity ZigBee device. Decoding at WiFi is a light-weight function that recycles the output from idle listening, thereby minimizing the computation while keeping compatibility to WiFi standard. SymBee is extensively evaluated both theoretically and experimentally through extensive testbed evaluations on six distinct indoor/outdoor locations under various degree of interference and noise. SymBee reaches the throughput of up to $31.25kbps$, $145.4\times$ faster than the state-of-the-art.

I. INTRODUCTION

Explosive growth of wireless devices over the last decade is anticipated to be intensified and diversified as we step into the Internet of Things (IoT) era, to reach 50 billion by 2020 [2]. As much as massive scale wireless body has enriched our daily lives, spectrum shortage has become one of the significant bottlenecks to efficient networking. I.e., overcrowded unlicensed ISM band has led to severe cross-technology interference (CTI) [12], which has become a major hurdle to network reliability and spectrum efficiency. For example, ZigBee is known to suffer from up to 50% packet loss under WiFi interference [21].

To address CTI, latest studies have introduced cross-technology communication (CTC) [7], [19], [36], [9], [38] that enable direct communication among heterogeneous wireless devices with different physical layers. CTC not only fundamentally resolves CTI via cross-technology channel coordination, but also serves as a building block to advanced services through collaboration beyond technologies. Specifically, they convey messages by modulating the timing [19], power [9], and duration [7] per packet basis. Although effective, they commonly suffer from limited data rate (215 bps for Zig-Bee $\rightarrow$ WiFi [34]) inherently imposed by the coarse-grained packet-level modulation. We note that there has been a recent advancement in fine-grained physical layer designs [16], [20]; However they are not applicable to ZigBee to WiFi scenario.

This work introduces SymBee, a novel symbol-level ZigBee to WiFi CTC reaching up to $31.25Kbps$, thereby improving packet-level predecessors by $145.4\times$. SymBee is uniquely built on the new insight on ZigBee-WiFi *cross-observability*. I.e., due to frequency overlap, ZigBee signal naturally flows into the WiFi RF front-end to be processed by the idle listening mechanism, where the output of the module illustrates how ZigBee signal is observed at WiFi. This reflects radio asymmetries between ZigBee and WiFi, including sampling rate/bandwidth and central frequencies. By carefully inspecting through the physical layer properties of ZigBee signal, SymBee is designed so that the combinations of ZigBee symbols (thus symbol-level) yield a special output pattern at the WiFi idle listening to maximize decoding reliability.

The highlight of SymBee design lies in its simplicity and compatibility – despite the underlying rationale involving in-depth details on physical layer signal, SymBee encoding turns out to be as simple as putting specific byte patterns in a payload of a legitimate ZigBee packet, which we call *payload encoding*. Specifically, the bytes are selected such that the corresponding ZigBee symbols generate intended output upon cross-observation at WiFi idle listening. This ensures full compatibility to the off-the-shelf ZigBee device and WiFi standard. Furthermore, since WiFi idle listening continuously runs by default (to detect any incoming WiFi packets), recycling its output amortizes much of the computational cost and enables light-weight decoding. This keeps the overhead and intrusion to the legacy communication minimal.

To summarize, SymBee is designed as a ZigBee to WiFi CTC in the aim to support upstream (or convergecast) which takes majority portion of IoT traffic, e.g., uploading sensing data. The contribution of this work is three-fold:

- To the best of our knowledge, for the first time, we analyze the physical layer cross-observability of ZigBee signal at WiFi RF front-end (via packet detection mechanism), both experimentally and analytically through rigorous derivations. Although the case of WiFi and ZigBee was investigated, the observation can be generally applied to understanding the interaction between disparate RF

IEEE computer society

front-ends, and can be extended to designing symbol-level CTC between other technologies.

- A novel ZigBee to WiFi CTC of SymBee is introduced. Payload encoding is as simple as customizing byte patterns in the payload of a legitimate ZigBee packet, which can be performed on any off-the-shelf devices. Decoding facilitates the default WiFi idle listening operation defined in the 802.11 standard, making SymBee a non-intrusive and energy-economic design. Simple preamble-based decoding enables robust communication under heavy noise.

- We evaluate SymBee both analytically and experimentally, where we implement the prototype on TelosB and USRP platforms. We extensively evaluate the performance under six different indoor and outdoor scenarios, where it is demonstrated that SymBee throughput reaches $31.25 kbps$, improving the state-of-the-art by more than $145.4\times$.

The rest of this paper is organized as follows. Section II introduces the motivation, where Section III provides a design overview. Section IV presents technical details of our design, followed by a robust technique and other notable features in Sections V and VI, respectively. Analytical and experimental performances of SymBee are in Sections VII and VIII. Related work are discussed in Section IX. Finally, Section X concludes the paper.

II. MOTIVATION

This section illustrates the values that SymBee would bring in diverse domains for wireless networking, followed by the need for symbol-level ZigBee to WiFi CTC.

A. Opportunities for CTC

With ever-increasing body of devices with incompatible heterogeneous wireless technologies, CTI has now become one of the major causes of network degradation [14], [37], [38]. This is even more severe for the low-power wireless techniques such as ZigBee, where it has been reported that up to 50% of the ZigBee packets are lost due to WiFi interference [21]. SymBee offers opportunity to mitigate the CTI and coordinate heterogeneous devices via, for example, (i) explicit coordination among IoT devices using cross-technology RTS/CTS instead of implicit CSMA/CA, (ii) cross-technology broadcasting of ZigBee spectrum occupancy to assist WiFi devices to switch to idle or less crowded channels. Such cross-technology channel coordination requires real-time and distributed message exchanges, which can be exclusively achieved by CTC. Additionally, CTC enables IoT devices to deliver data (e.g., sensing info.) directly to WiFi (i.e., to the Internet and cloud), subtantially enhancing IoT in various aspects including connectivity, accessibility, and responsiveness.

B. The Need for Symbol-level CTC

Limitation of Gateway. Traditionally solution for communication between multiple technologies has been through mult-radio gateways. However, deployment of gateways impose several practical issues (especially compared to CTC) such as: (i) increase in communication hops, leading to not only more delay, and more importantly, more traffic (flowing into and out from gateway). This further aggravates competition in ISM spectrum, (ii) increase in network deployment complexity, and (iii) the lack of mobility support.

Limitations of the State-of-the-art. A stream of CTC designs in literature take packet-level approaches [7], [9], [19], [38], where they use the *packet* as the basic unit in modulation (analogous to 'pulse' in physical layer) – E.g., [9] uses power of each packet to modulate CTC message. While packet-level designs are simple to adopt and are highly compatible with the legacy devices, they inherently suffer from bounded bandwidth, or throughput. For example, the state-of-the-art ZigBee to WiFi CTC reports the throughput of $215 bps$ [34], limiting the usage to delivering short control information. A recent line of CTC studies take physical-layer approaches, where signal from one wireless device closely emulates the waveform of the other [16], [20]. Despite their vastly enhanced speeds, they are inherently inapplicable to ZigBee to WiFi CTC due to a large bandwidth gap (2 vs 20MHz) between the technologies – essentially limiting the ZigBee signal's degree of freedom to fall short (for emulating WiFi).

Advantages and Challenges of Symbol-level CTC. This calls for a new method (i.e., Symbol-level approach) for the breakthrough in the rate, and to expand the practical use of ZigBee to WiFi CTC. Considering duration of the minimal ZigBee packet of 576 us (i.e., 18 bytes), the bandwidth of packet-level CTC becomes 1.736KHz ($=\frac{1}{576us}$) . Conversely, exploring physical layer for symbol ($16\mu s$) level CTC expands the bandwidth to $62.5 KHz$. By Shannon's law, the theoretical bound of the data rate is linear to the bandwidth, therefore, symbol-level approach can vastly improve the throughput of the packet-level approach by $36\times$. The challenge of symbol-level CTC is in maintaining the compatibility to the legacy devices. This is because symbols are strictly defined in the physical layer where their manipulation could easily lead to standard incompatibility, if not carefully designed.

III. DESIGN OVERVIEW AND BACKGROUND

This section discusses the overview followed by technical background of our design design.

A. SymBee in a Nutshell

SymBee is a ZigBee to WiFi CTC technique that vastly improves the data rate of the state-of-the-art designs by exploiting physical layer signatures. This is effectively achieved by the two unique features: (i) ZigBee's physical layer signature is indirectly controlled by manipulating the payload bytes, which we call payload encoding, such that (ii) the payload exposes intended (i.e., message-bearing) patterns when the it is processed by the WiFi's idle listening mechanism. In other words, SymBee is carefully designed not only considering the physical layer properties of ZigBee and WiFi, but also

501

their *cross-observability* – i.e., how ZigBee signal is processed when fed into WiFi's idle listening.
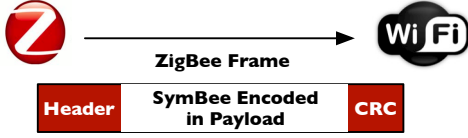

Fig. 1. SymBee message is embedded within ZigBee packet, simply by encoding payload with specific byte patterns.

SymBee design is extremely light-weight and fully compatible to standards, making it nondistruptive to ZigBee and WiFi operations. Figure 1 illustrates how SymBee message is embedded into ZigBee packet payload. Encoding at the transmitter (i.e., ZigBee) is as simple as selecting byte patterns of the payload, which does not require any hardware/firmware change to commodity ZigBee platform. Decoding at WiFi receiver recycles the computational result of the idle listening which runs continuously by default, thereby minimizing the computation cost while maintaining compatibility to the WiFi standard.
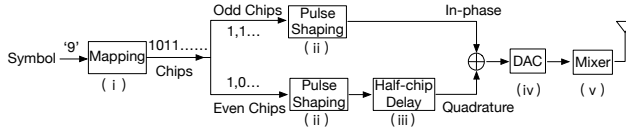

Fig. 2. ZigBee transmitter

symbol	chips sequence
0	11011001110000110101001000101110
.	.
.	.
.	.
F	11001001011000000111011110111000

TABLE I
ZIGBEE (802.15.4) SYMBOL TO CHIP SEQUENCE MAPPING

B. ZigBee-WiFi Cross-observability

We discusses how ZigBee signal is cross-observed at the WiFi idle listening, which serves as the basis of our design.

ZigBee transmitter. Figure 2 illustrates the ZigBee transmitter running Offset Quadrature Phase-Shift Keying (OQPSK) modulation, which covers the entire conversion procedure from input symbol to output signal wave. Initially, data to be transmitted is cut in units of 4bits called symbol (thus ranging 0-F). Then, in step (i), each symbol is mapped to unique 32 chip sequences as per Table I – a technique called Direct Sequence Spread Spectrum (DSSS).
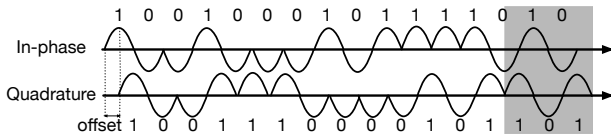

Fig. 3. ZigBee baseband signal corresponding to symbol 6. As marked in gray, a symbol may have parts with coninuous sinusoidal waves in both in-phase and quadrature, which can easily be cross-observed at WiFi.

In step (ii), 32 chips are divided into odd and even chips where chip 0 and 1 are converted to positive and negative half-sine pulses, respectively. Pulses generated from odd chips are referred to as in-phase signal. On the other hand, in (iv) pulses from even chips are delayed by half pulse duration of $0.5us$, and are referred to as Quadrature signal. Figure 3 depicts an example of in-phase and quadrature (i.e., I/Q) signals corresponding to symbol 6. Gray indicates both in-phase and quadrature are continuous sinusoidal, which can easily be *cross-observed* at WiFi (details in the following parts of the section). I/Q signals are merged and in step (iv), converted to analog continuous waveform via DAC. Finally, in (v) the mixer shifts the baseband signal to the carrier frequency (i.e., passband) which is then pushed to the antenna for transmission.
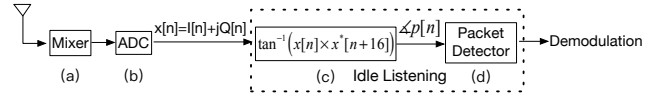

Fig. 4. WiFi receiver. ZigBee signal flows into the WiFi idle listening where it is cross-observed.

WiFi receiver. To provide insight on cross-observability, Figure 4 illustrates WiFi receiver operation up to idle listening[1]. Frequency overlap with WiFi naturally lead ZigBee signal to flow into WiFi RF front-end. Then, in (a) mixer shifts this signal to baseband, where in (b) it is sampled to digital domain at 20Msps (i.e., Nyquist rate) or above. The samples are fed into the idle listening. In search for WiFi packet, idle listening continuously processes any incoming signal including ZigBee. This is done in (c) by computing the phase differences between 16 samples denoted as $\angle p[n]$. WiFi signal is prepended with Short Training Sequence (STS), which is a sequence of repeated signals with interval of 16 samples (i.e., $0.8\mu s$) for the total duration 160 samples. In other words, (d) detects WiFi packet only when $\angle p[n] = 0$ holds for STS duration, which then passes the signal for demodulation.
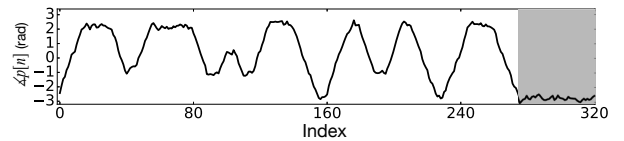

Fig. 5. Cross-observation of ZigBee symbol 6.

When ZigBee signal is fed into idle listening, (c) yields corresponding $\angle p[n]$ pattern of the signal, which is the cross-observation of the ZigBee from WiFi. Figure 5 depicts an example for ZigBee symbol 6 obtained from our experiment[2]. With the mathematical derivation of phase (i.e., $\angle p[n]$) in Appendix A, the figure clearly demonstrates that cross-observation outputs notable patterns where the phase is kept

---

[1]Figure 4(c) is a part of autocorrelation-based idle listening. Such design (e.g., Schmidl-Cox) is widely used in practice for its low complexity and robustness.

[2]For the sake of clarity, we assume the channel frequency offset between WiFi and ZigBee has been compensated, where this is simply adding $\frac{4\pi}{5}$ to the phase difference. Detailed discussion and derivation is given in Appendix B.

stable in the gray region. This corresponds to the gray portion in Figure 3, where in-phase and quadrature signals are both continuous sinusoidal waves, indicating that such stable phases are easily detectable with minimal computational overhead (details in Section IV-C). SymBee exploits this unique feature in cross-observation to achieve reliable and light-weight CTC. To sum up, by (i) exploring the intrinsic properties of ZigBee symbols as defined in the ZigBee standard, and (ii) recycling the phase values computed by default in WiFi, SymBee remains fully compatible to both standards and non-disruptive to their operation.

IV. SymBee Design

This section provides technical details and insights on SymBee.

A. Robust Payload Encoding@ZigBee

SymBee's payload encoding is built on top of the observation on the stable phase (i.e., $\angle p[n]$), where we design a technique to maximize robustness. SymBee essentially selects optimal combinations of ZigBee symbols such that (i) they yield the *longest* possible stable phase that maximizes detection under noise and interference, and at the same time, (ii) the phase of different combinations are maximally distinct, which minimizes decoding errors. The combinations are assembled from the 16 (i.e., 0-F) symbols defined in the ZigBee standard (Table I), thereby optimizing the performance while maintaining full compatibility to tens of millions of commercial ZigBee devices.

Specifically, ZigBee symbol combinations of 6 and 7 are used to convey SymBee bit 0 while E and F represent bit 1. For simplicity, the combinations are denoted as (6,7) and (E,F) thereafter. Given the symbol combinations, SymBee's payload encoding is as simple as converting bits to be transmitted, to either (6,7) and (E,F). We note that since a symbol is worth 4 bits, combination of two symbols is expressed as a single byte. In other words, every SymBee bit is represented as a byte put into the payload of a ZigBee packet, thus encoding payload. In the following, we study the uniqueness and the optimality of the symbol combinations (6,7) and (E,F) in terms of communication robustness (testbed-evaluated against noise and interference in Section VIII).

(a) ZigBee symbol combination (6,7)
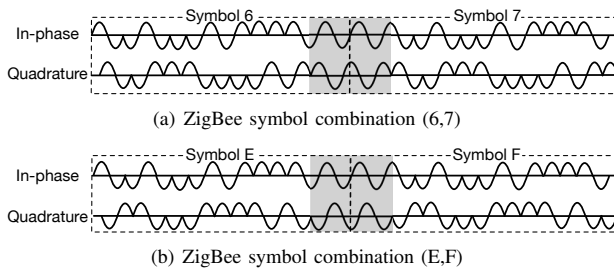
(b) ZigBee symbol combination (E,F)

Fig. 6. Concatenated ZigBee symbols (6,7) and (E,F) represent SymBee bits 0 and 1, respectively. Gray indicates continuous sinusoidal region, maximized by concatenating two symbols. The two combinations yield the longest stable phase among any other combinations of arbitrary number of symbols.

Longest Stable Phase. To maximize robustness to noise/interference, symbol combinations with the longest stable phase are found through a careful analysis on chip sequences of symbols – such that serially concatenating them yields the longest continuous sinusoidal wave at the junction. Figure 6 illustrates the two combinations with the longest stable phase, (6,7) and (E,F) representing SymBee bits 1 and 0, respectively. Parts with continuous sinusoidal I/Q (cross-observed as stable phase) are marked in gray, which are $5\mu s$ long. This is reflected as $4.2\mu s$-long stable phases at cross-observing WiFi ($0.8\mu s$ difference corresponds to 16 samples) for both (6,7) and (E,F). The two combinations yield the longest stable phase among any other combinations with arbitrary number of symbols, indicating that they are indeed the optimal selection for maximum robustness to noise/interference.
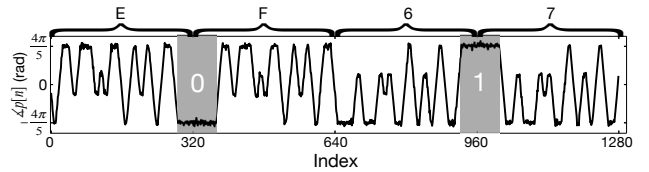

Fig. 7. Phase ($\angle p[n]$) of SymBee signal in practice, when bits 0 and 1 are sent back to back. Gray region indicate corresponding stable phase.

Optimized SymBee Bit Distinction. The symbol combinations of (6,7) and (E,F) has another powerful characteristic that optimizes them for robust decoding. Figure 7 demonstrates the actual $\angle p[n]$ values when the SymBee bits 0 and 1 are sequentially transmitted, where the stable phase corresponding to the SymBee bits are in gray. As noticeable in the figure, stable phases indicating 0 and 1 are $\mp\frac{4\pi}{5}$, respectively, which correspond to the minimum and maximum among all possible phase values in the cross-observation (derivation in the section IV-B). In summary, (6,7) and (E,F) are optimized to decoding, with maximum possible distinction between 0 and 1 (i.e., $\frac{8\pi}{5}$).
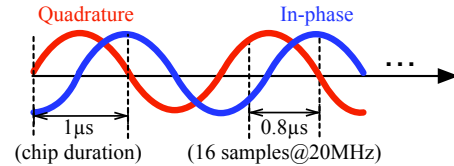

Fig. 8. Long continuous sinusoidal signal generated by (6,7). With $1\mu s$ chip duration, both in-phase and quadrature yield sinusoids of 0.5MHz where half-chip offset ($0.5\mu s$) between them indicates $\pi/2$ phase difference.

B. Understanding Stable Phase

This section provides conceptual description of the stable $\angle p[n]$, followed by a mathematical derivation of the value. Figure 8 shows the continuous sinusoidal signal generated by (6,7) (i.e., gray area in Figure 6(a).) as well as the $\pi/2$ phase difference between in-phase and quadrature signal caused by half-chip ($0.5\mu s$) offset. The stable phase indicates that such a signal, when fed into WiFi idle listening, yields consistent phase difference. Specifically, $\angle p[n] = \frac{4\pi}{5}$ between 16 sam-

503

ples at 20Msps. Derivation of the stable phase difference is as follows.

Derivation. As depicted in Figure 8, $1\mu s$ half-sine chip indicates the frequency of the continuous sinusoid is 0.5MHz. The figure also shows $\frac{\pi}{2}$ phase difference between In-phase and Quadrature continuous sinusoidal; therefore they can effectively be presented as $-\cos(2\pi\frac{10^6}{2}t)$ and $\sin(2\pi\frac{10^6}{2}t)$ respectively, or $-e^{-j2\pi\frac{10^6}{2}t}$ in complex representation. Recall that, from Figure 4(c), $\angle p[n]$ is computed by the WiFi idle listening as:

$$\angle p[n] = tan^{-1}(x[n] \times x^*[n+16]) \tag{1}$$

where $x[n]$ is n-th sample and $x^*[n+16]$ is a complex conjugate of the sample $n+16$. That is, $\angle p[n]$ essentially indicates the phase difference between the two samples, $x[n]$ and $x[n+16]$, which are 16 samples apart. Under sampling rate of 20MHz, 16 samples interval represents $0.8\mu s$ in time, which can be directly plugged in to t of $-e^{-j2\pi\frac{10^6}{2}t}$ (i.e., complex representation of I/Q signals in (6,7)) to obtain $\angle p[n] = \frac{4\pi}{5}$. Similarly, $\angle p[n] = -\frac{4\pi}{5}$ for continuous sinusoidal within (E,F), as its I/Q signal is precisely the conjugate of the continuous sinusoidal in (6,7) as can be observed in Figure 6. We note that the stable phase difference of $\pm\frac{4\pi}{5}$ induced by (6,7) and (E,F) are kept consistent for $4.2\mu s$ until in-phase and/or quadrature becomes discontinuous, providing the longest stable phase among all ZigBee symbol combinations and optimal bit distinction with maximum and minimum $\angle p[n]$ among all 17 possibilities derived in Appendix A.

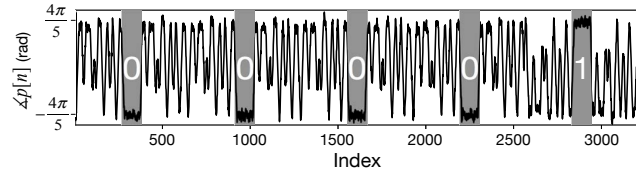### C. Extremely Light-weight Decoding@WiFi

Fig. 9. SymBee preamble is essentially four consecutive 0's, prepended to SymBee message.

The use of the stable phases enables extremely light-weight decoding. This is essentially done by checking the signs of phase values, where the decision boundary of 0 minimizes the error (i.e., negative $\leftrightarrow$ nonnegative) under random noise. More specifically, since the stable phases is $4.2\mu s$ long consisting of 84 phase values, decoding is simply monitoring consecutive 84 phase values if they are consistently kept negative or nonnegative (i.e., below or above the decision boundary of 0), which indicate SymBee bit 0 or 1. In practice, phase values suffer from noise and up to threshold (τ) number of errors are tolerated. In our experiment τ is set to be 10 where both false positive and negative are kept under 3% at SNR as low as -5dB, a harsh SNR for most scenarios [31]. To sum up, at any time t, SymBee decoding is performed by monitoring the phase values with the window size of 84 – from $\angle p[n]$ to $\angle p[n+83]$. SymBee bit 1 or 0 is detected whenever the

window has more than $84-\tau$ negative or nonnegative values, respectively. In all other cases, SymBee bit is not detected.

V. ENHANCED DECODING WITH SYMBEE PREAMBLE

Here we introduce an optional technique that further improves the resilience to noise, by prepending a simple SymBee preamble. SymBee bit consists of a pair of ZigBee symbols(i.e., (6,7) or (E,F)), or in other words, SymBee bits repeat periodically with every two ZigBee symbols, corresponding to 640 samples (=32us). This indicates that the stable phases (i.e., the actual regions containing SymBee bits) is only 84 samples (=4.2us) out of 640, where other parts need not be considered in SymBee decoding. SymBee preamble essentially explores this property to substantially reduce the chance of decoding error. SymBee preamble, which is simply four consecutive SymBee bit 0, enables precise detection of the bit start time, such that only the part holding SymBee bit is considered in the decoding, effectively filtering out non-contributing portions.
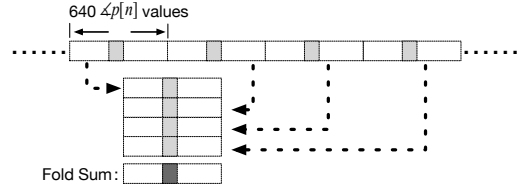

Fig. 10. Illustration of Folding. Gray indicate portions with stable phase.

Capturing SymBee Preamble. Figure 9 illustrates SymBee preamble, or four consecutive 0's leading SymBee message. The preamble can be effectively and reliably captured via *folding* – a technique introduced in [30] to detect periodic signal under noise, which in our case, is the four repeated stable phases due to four (E,F) (i.e., SymBee bit 0). The process of folding is illustrated in Figure 10: One (E,F) takes up 640 values (32 μs), and hence, the length of the SymBee preamble is $640 \times 4 = 2560$ values. This vector of 2560 phase values are sliced into four subvectors of size 640 and stacked up (i.e., folded) to form a 4×640 matrix. Then, each columns are summed to yield a vector of size 640, which we refer to as *Fold Sum*. In other words, *Fold sum* $= \sum_{i=0}^{3} \angle p[n+640i]$ where $0 \leq n < 640$. Then, capturing preamble is achieved by applying the decoding (described in Section IV-C) to the fold sum.
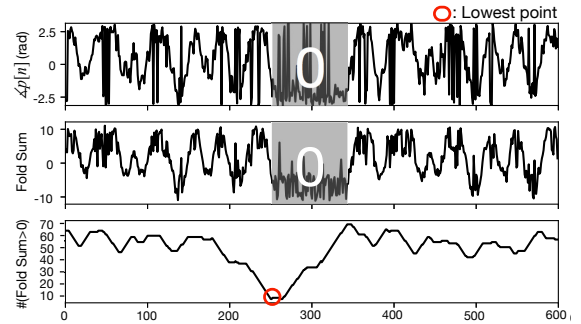

Fig. 11. SymBee preamble detection success example (SNR=-10dB)

Figure 11 demonstrates an example of preamble capturing in practice. The figure shows a scenario of poor signal quality (SNR = -10dB), where $\angle p[n]$ is very unstable and thus standard decoding is infeasible (top figure). Under the harsh environment, folding stabilizes the stable phase values (middle figure) and enables SymBee preamble to be safely captured (bottom figure) thereby significantly enhancing robustness. We also note that preamble can be further protected by increasing the repetitions, where four offered reliable ($> 98.7\%$) capturing in our experiments under low SNR of -10dB.

Decoding under synchronized bit timing. Capturing preamble enables precisely locating the SymBee bits for decoding. This significantly decreases the chance of error, by skipping regions that does not bear the SymBee bits. Locating the ZigBee bit is straight-forward; Upon capturing preamble, the receiver marks the index of the initial phase value within the captured preamble. Suppose the index is n_0, then the first symbol (i.e., initial part of SymBee data) starts at $n_1 = n_0 + (640 \times 4)$, which takes the preamble length into account. The following parts of the SymBee data occurs every 640, i.e., $n_2 = n_1 + 640$, $n_3 = t_2 + 640$, and so on, until reaching the end of the SymBee message. Decoding is performed on 84 samples starting at each bit location (e.g., n_k to $n_k + 83$ for k-th SymBee bits), where the samples outside the range are ignored as they do not embed any SymBee bit. Since the position of potential SymBee bits are already located, we set $\tau_{sync} = 42$ (i.e., half of the stable phase) to decide the SymBee bits: out of 84 values in stable phases, τ_{sync} or more above 0 indicates SymBee bit 1, otherwise, 0 – essentially turning decoding to majority voting for higher noise tolerance ($cf.$ Section IV-C).

VI. SymBee Features

This section discusses unique features and simple extensions that enable a boader applicability.

A. WiFi-ZigBee Cross-technology Broadcast

SymBee message is embedded in a normal ZigBee packet, only with specific payload. Therefore, the same SymBee message can naturally be received by a ZigBee node simultaneously as being delivered to WiFi – i.e., SymBee is capable of transmitting cross-technology broadcast to both WiFi and ZigBee. SymBee message reception at the ZigBee is done in two simple steps: First, SymBee preamble is captured with four consecutive bytes of 0x67 (i.e., symbols (6,7)), corresponding to four SymBee bits of '0'. Then, following bytes of 0x67 or 0xEF are interpreted as bit '0' or '1', respectively. We note that this can be done at the application code on any standard ZigBee device, without any change to the firmware. Such cross-technology broadcast may serve as a key enabler to various services, including explicit channel access control between WiFi and ZigBee. For example, a SymBee message may include the time/frequency allocation for ZigBee, which is notified to WiFi (to restrain channel usage) and ZigBee (to promote channel usage) at the same time. This would yield precise, efficient, and immediate channel utilization;

that is, without the inevitable inefficiency and potential risk of interference that typical implicit, contention-based channel access (e.g., CSMA/CA) mechanisms commonly suffer from.

B. Compatibility to 40MHz WiFi

Technical descriptions throughout the paper focuses on the widely deployed WiFi with 20MHz bandwidths (e.g., 802.11g/n), only for the sake of clarity. SymBee is in fact fully compatible to 40MHz 802.11n WiFi, with the sender side (i.e., ZigBee) kept identical. The receiver side (i.e., WiFi) is simply scaled to cope with the doubled sampling rate, which enhanced the decoding reliability. Specifically, the $\angle p[n]$ is computed as $\angle p[n] = \angle(x[n]x^*[n+32])$ as per twice the sampling rate ($cf.$ Eq. 1). The stable phase values remain $\pm\frac{4\pi}{5}$, while the number of stable phase values is doubled to 168 ($= 84 \times 2$). To locate SymBee bits, $640 \times 4 \times 2 = 5120$ phase values should be skipped following the SymBee preamble. At the decoding stage, the interval between two SymBee bits is 1280 at 40MHz WiFi receiver, as opposed to 640 under 20MHz WiFi. Finally, 84 steady phase values above 0 (i.e., the decision boundary) out of 168 indicate SymBee bit '1', and '0' otherwise. Overall, doubled stable phase values improves the robustness with the capacity to tolerate twice the errors.
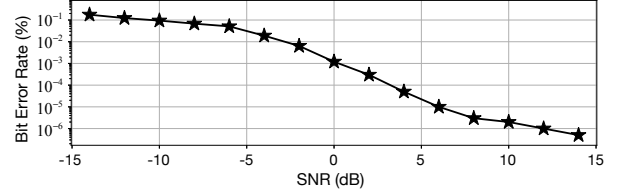

Fig. 12. Numerical Result of Bit Error Rate under different SNR.

VII. Analytics

This section offers analysis on Bit Error Rate (BER) of SymBee, followed by the bitrate. BER is computed with respect to SNR;. Intuitively, low SNR (high noise) leads to phase value to fluctuate out of decision boundary to cause decoding error. We use Pr_ϵ to denote the the probability of error in phase value (i.e., crossing the boundary). More specifically, this is when the phase value of SymBee bit 0 or 1 is higher or lower than the decision boundary, respectively. This yields $Pr_\epsilon = Pr(\angle p[n] > 0|bit = 0) = Pr(\angle p[n] < 0|bit = 1)$ under random noise. Since SymBee bit is decoded following the majority voting, BER is computed as:

$$BER = \sum_{l=42}^{84} C_{84}^l Pr_\epsilon^l (1 - Pr_\epsilon)^{84-l} \quad (2)$$

where the distribution of Pr_ϵ under different SNR is obtained from widely-used GNURadio. As shown in Figure 12, BER of SymBee is lower than 10% even under SNR of -10dB. On the other hand, ZigBee throughput can be found via straightforward computation: Since SymBee transmits 1 bit per two ZigBee symbols while ZigBee delivers 4 bits per symbol. This yields $\frac{1}{8}$ the bitrate compared to ZigBee, i.e., 31.25kbps.
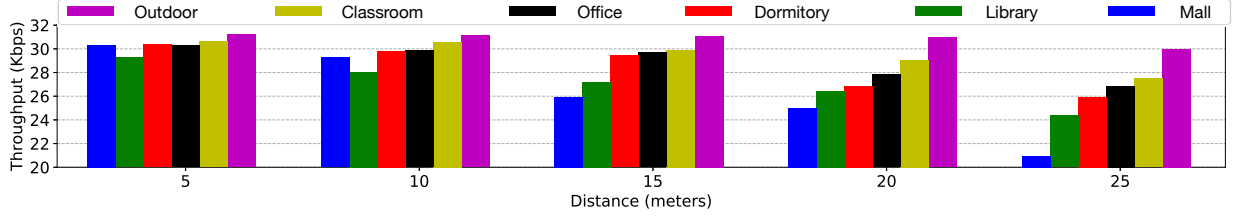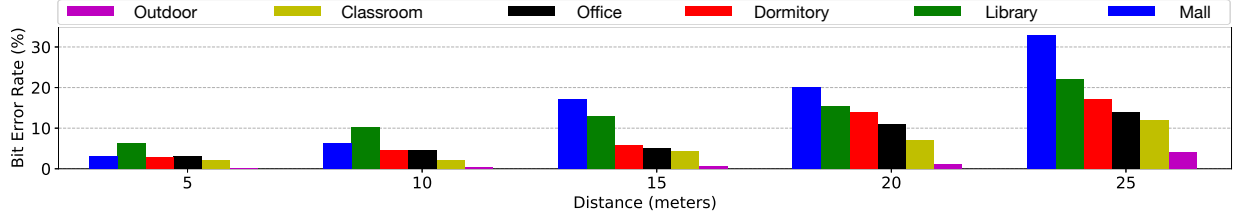
505

Fig. 13. Throughput of SymBee in 6 scenarios.

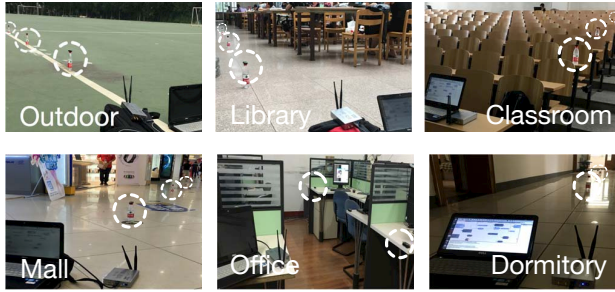Fig. 14. Bit Error Rate of SymBee in 6 scenarios.

Fig. 15. We evaluate the performance of SymBee in six representative areas: outdoor, library, classroom, dormitory, office and mall.

VIII. EVALUATION

We implement SymBee prototype on TelosB and USRP B210 with GNURadio 3.7.9 [3] and evaluate them in six representative areas of outdoor, library, classroom, dormitory, office and mall, as illustrated in Figure 15. We set the maximum payload to 127 including 2 bytes control information, 1 byte data sequence and 2 bytes check sum. WiFi idle listening has been implemented as a built-in block in GNURadio. We extract phase information from this block and implement SymBee receiver on USRP B210. We also note that SymBee can be implemented in 802.11 compliant platforms such as WARP [1] with minimum code modification.

A. Throughput

We present the effectiveness of our design by evaluating throughput under six scenarios in Figure 15, at distance of 5∼25 meters. As shown in Figure 13, 31.25Kbps can be achieved within 15 meters while the throughput of SymBee still remains at 30Kbps at the distance of 25 meters in the outdoor scenario. Since there are no cross-technology interference and obstacles, the throughput of SymBee in outdoor scenario is higher than other scenarios. The throughput of classroom, ≥ 27.5Kbps within 25 meters, is the 2nd highest among 6 scenarios. In the dormitory environment mild WiFi traffic was occurring during the experiment, which caused mild interference to SymBee. In the office and dormitory,

most computers are connected through high speed wire cables. However, since the number of private WiFi access points and users in office is less than in dormitory, SymBee in office achieves ≥ 26.9Kbps within 25 meters, which is higher than ≥ 25.8Kbps in the dormitory.

As shown in Figure 13, SymBee only achieves ≥ 21Kbps within 25 meters in the mall due to the signal blockage from shoppers and a large amount of private WiFi access points in the stores. In the library, almost all students are connected to campus WiFi via laptops or smartphones, causing significant WiFi interference. Therefore the throughput is lower than other scenarios. The throughput of SymBee in the mall and library achieve only ≥ 21 and ≥ 24.4Kbps within 25 meters, respectively.
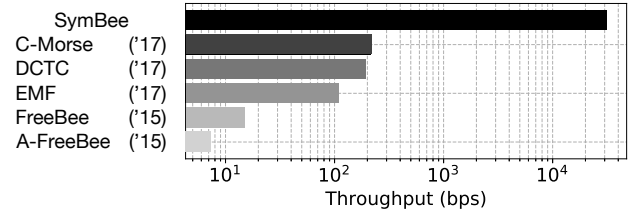

Fig. 16. Comparison with other CTC approaches in the same setting.

We also compare SymBee with FreeBee[19], A-FreeBee[19], EMF[8], DCTC[15], C-Morse[34], 5 cross-technology techniques supporting ZigBee to WiFi communication, in the same setting. In our experiment, ZigBee senders send out 100 packets with 50 repeated SymBee bits '01' per second with maximum transmission power (0 dBm). Throughput of C-morse is 215bps[34] when distance between sender and receiver is 1.5 meters in the office scenario. Figure.16 indicates SymBee outperforms C-Morse, the state-of-the-art ZigBee to WiFi cross-technology approach, by 145.4×.

B. Bit Error Rate

We present robustness of our design by evaluating bit error rate (BER). The BER of 6 representative scenarios are

506

presented in Figure 14. We can see the trend of BER in 6 areas clearly. SymBee reaches lowest BER, i.e. $\leq 5\%$, regardless of distance in the outdoor, showing that SymBee is robust enough to resist severe noise. For the indoor environments, SymBee achieves $\leq 10\%$ bit error rate within 10 meters even in the crowded mall and library.
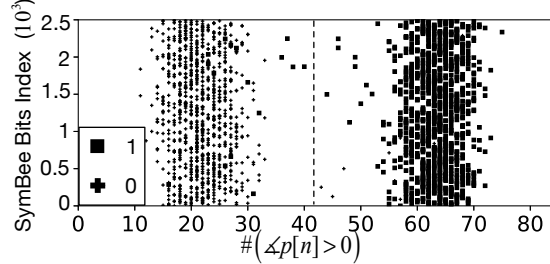

Fig. 17. Constellation diagram: we record the number of stable phase values which are above 0 for each SymBee bit in outdoor scenario, where SymBee sender and receiver are deployed 15 meters away.

Figure 17 shows the constellation diagram of outdoor scenario along with the decoded SymBee bits when the 2 bits of '01' is sent 2500 times. The x-axis of this figure indicates the number of stable phases above the decision boundary for each SymBee symbol. Decoding is successful when SymBee bit 1 (blue square) resides inside the right part and SymBee bit 0 (red square) resides inside the left part of the constellation diagram. Figure 17 depicts the distribution of the dots, in which $\geq 98\%$ are successfully decoded.
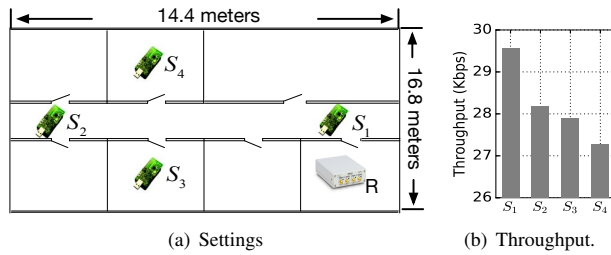

(a) Settings (b) Throughput.
Fig. 18. None Line-of Sight settings: we deploy ZigBee senders at 4 positions in the office.

C. None Line of Sight Scenario

Performance of NLOS setting is tested in office environment where we deploy ZigBee nodes at corridor and separate rooms as shown in Figure 18(a). In this evaluation, 4 ZigBee senders working with maximum Tx power on ZigBee channel 13 are placed at $S_1 \sim S_4$ and WiFi receiver is placed at R. The throughput of $S_1 \sim S_4$ are 29.5, 28.2, 27.9 and 27.3 Kbps respectively. Since S_1 is the closest to R, throughput of S_1 is highest among 4 nodes. Even though S_3 is closer to R than S_2, throughput of S_3 is lower than S_2 due to more blockages from walls. This indicates the walls decreases the throughput of SymBee severely along with the distance between sender and receiver.

D. Impact of Transmission Power

We investigate the impact of transmission power on SymBee. Transmission power of a ZigBee node affects its coverage. Different TX power yields different multi-path reflections
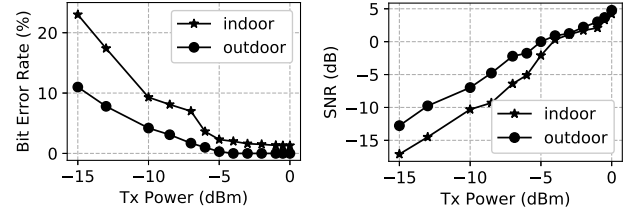
(a) BER under different power level. (b) SNR under different power level.
Fig. 19. Impact of transmission power.

and fading. We set a TelosB node to different Tx power ($-15 \sim 0\text{dBm}$) and deploy it 5 meters away from WiFi receiver in the office at midnight and outdoor as a comparison.

As shown in Figure.19(a), SymBee reaches BER $\leq 10\%$ within -10dBm and $\leq 23\%$ within -15dBm. As demonstrated in Figure 19, SNR of same TX power in the indoor environment is lower than outdoor, thus resulting in higher BER. This is because multi-path effect in indoor environment caused by the blockage and bounce of walls is much more severe than outdoor environment.
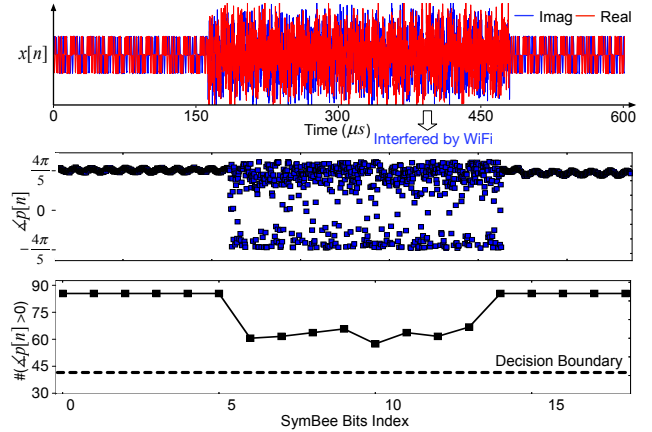

Fig. 20. An example of WiFi interfered SymBee signal and the corresponding number of phase values in the boundary.

E. Robustness to Interference

The ubiquitous WiFi interference is a major reason of ZigBee packet corruption. At first, we obverse that SymBee bits could always be decoded correctly even from the severe interfered signal. Figure 20 shows a segment of SymBee packet, where all SymBee bits are '1's, is interfered by a 270 μs WiFi signal. The signal to interference plus noise ratio is 0dB indicating that the WiFi is as strong as SymBee signal. The stable phase values is ideally 84 samples long while under interference it drops to approximate 60; but being still larger than 42, (i.e., the half of ideal length) it is correctly decoded. Thus, this SymBee packet is robust enough to overcome the 0dB interference.

To further verify the robustness of SymBee under different interference level, we conduct a trace driven experiment based on the pure SymBee signal and WiFi 802.11g signal we collect on USRP B210. Mixed with different power level WiFi signal,
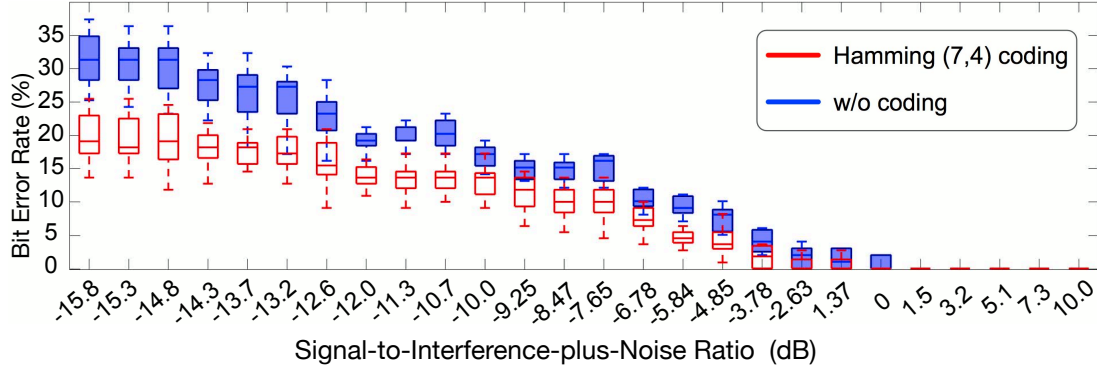
Fig. 21. Comparison of Bit error rate when no coding and Hamming (7,4) coding is applied.

Bit error rate of SymBee are represented by blue boxes in the figure 21. BER turns to be 19.5% when SINR drops to -10dB, meaning that the strength of WiFi interference is 2 times of SymBee signal. Even though the BER under strong interference is high, frame reception ratio could be increased via link layer coding. By applying Hamming (7,4) link layer coding on top of SymBee, BER of SymBee with coding decreases to almost half of SymBee without coding. Even though Hamming (7,4) coding can only correct one bit out of 7 bits, this experiment shows the big potential of SymBee in terms of robustness.
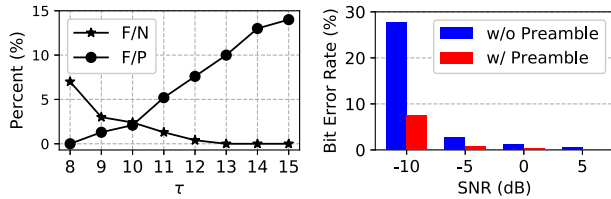
(a) Performance of SymBee with different τ under SNR of -5dB.

(b) Performance of SymBee with or without preamble.

Fig. 22. Impact of τ and preamble.

F. Impact of τ and preamble

We show the how τ affects detection of SymBee bits under SNR of -5dB in Figure 22(a). Higher τ indicates less SymBee bits would be missed while the false positive (F/P) ratio is getting higher. Therefore, we set τ to 10 where both false positive and false negative (F/N) are well balanced at a reasonably low values. Figure 22(b) depicts the bit error rate (BER) with and without preamble. Under the SNR of -5dB, the BER of SymBee without preamble achieves 27.4%, where it drops to 7.6% with preamble. The significant enhancement of SymBee via prepending preamble is clearly shown in this figure.

G. Mobility

The mobile scenario is also taken into account in our experiments. We evaluate SymBee on a track&field as shown in Figure 23(a). We deploy a WiFi receiver (a laptop with USRP B210) on a track&field where ZigBee senders (TelosB nodes) pass by the receiver at different speed: walking (3.4

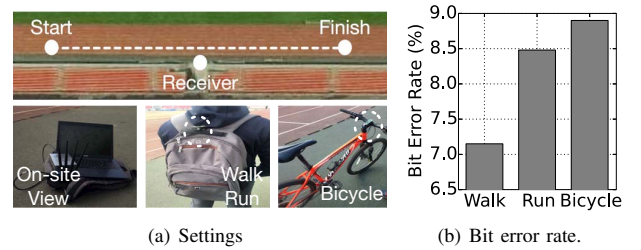(a) Settings

(b) Bit error rate.

Fig. 23. Mobile scenario evaluation: we evaluate the impact of mobility on SymBee on a track&field.

mph), running (5.3 mph) and riding a bicycle (9.3 mph). The BER of different speed, tested by 3 ZigBee senders, are 7.15%, 8.48% and 8.9% respectively as shown in Figure 23(b). The blockage and vibration of bag, physical body and bicycle mainly cause the distortion of ZigBee signal and received error bits. Therefore the BER of this mobile experiment is higher than outdoor scenario in Figure 15.

IX. RELATED WORK

This work lies in the intersection of three areas: cross-technology communication, interference mitigation, and heterogeneous collaboration.

Cross-technology communication. CTC was introduced to enable direction communication without the need for the gateway [4]. Doing so not only is beneficial in terms of cost savings, but also is advantageous in network planning as it largely simplifies network structure complexity, and enhances spectrum efficiency by removing the traffic running into and out of the gateway [28]. Most of the CTC work take the packet-level approach where a packet serves as the unit of modulation (similar to pulse in digital communication systems): Esense [7] and HoWiEs [38] modulates length of a single or sequence of packets, while FreeBee [19] modulates via beacon timings. B2W2 [9] delivers messages from Bluetooth to WiFi by controlling the power of Bluetooth packets. SymBee takes a unique approach of symbol-level CTC for throughput breakthrough. GapSense [37] introduces a fine-grained physical layer design, where it requires a special hardware. The latest work of WEBee [20] and BlueBee [16],

508

with the physical layer approach and high-throughput, are most similar, but are infeasible for ZigBee to WiFi communication.

Interference Mitigation. There had been much effort in the networking and wireless community to resolve interference under a wide range of scenarios and systems [11][23], where they can be roughly divided into PHY and link layer approaches. The former includes MIMO [25] [35] and OFDM [24] techniques. These techniques can be further categorized to interference cancellation [13], [27], [40] and interference alignment [5]. Another stream of work in interference mitigation is link layer designs [6] [29], such as enhancing packet robustness and recovering from errors [33]. Our design is fundamentally different from these work, where SymBee aims at achieving coordination via direction communication.

Heterogeneous collaboration. Connection and interoperation between heterogeneous wireless systems is traditionally established via gateways equipped with multiple radio interfaces to perform translation tasks [10] [18] [22]. Based on this structure, many systems that enables heterogeneous collaborations to explore the synergistic effect have been introduced [17], [26], [32], [39]. In WiZi-Cloud [17], for example, offers energy efficient Internet connectivity to mobile devices by utilizing ZigBee device with the access to the Internet via gateway.

X. CONCLUSION

We propose SymBee, a cross-technology communication framework that aims to bridge capacity and compatibility by customizing ZigBee packets. SymBee's encoding is as simple as putting specific byte patterns in the ZigBee packet payload, maximizing its applicability. This generates pattern at the PHY layer that can easily be detected at the WiFi idle listening. Theoretical analysis and extensive testbed experiments on TelosB nodes and USRP B210 reveal that SymBee is a reliable and efficient under various practical settings with the throughput up to 31.25Kbps, $145.4\times$ of the state-of-the-art.

APPENDIX A
PHASE DIFFERENCE COMPUTATION

Here we derive phase difference ($\angle p[n]$), obtained via cross-observation. We first note that quadrature and in-phase can be expressed as a sine and cosine functions, respectively, due to half-chip offset. Then, below two cases should be considered, depending on the positions of $x[n]$ and $x[n+16]$ in chips.

Case#1: When quadrature and in-phase of $x[n]$ and $x[n+16]$ are located in different chips, the two samples can be written as:

$$x[n] = e^{j\alpha_1(\omega n + \beta_1 \pi)}$$
$$x[n+16] = e^{j\alpha_2(\omega(n+16) + \beta_2 \pi)}$$
(3)

where $\omega = \frac{2\pi}{T_s}$, $\alpha \in \{1, -1\}$, $\beta \in \{0, 1\}$ and $n \in [10L + 4, 10L + 10), L \in N^+$. We note that the equations utilize the fact that half-sine ZigBee chips can be expressed with sinusoidal functions and their phase shifts. Then, by taking the phase difference from the equation we have:

$$\angle p[n] = (\alpha_1 - \alpha_2)n\omega - 16\alpha_2\omega + (\alpha_1\beta_1 - \alpha_2\beta_2)\pi \quad (4)$$

Case#2: When quadrature or in-phase of $x[n]$ and $x[n+16]$ are located in the same chip, the two samples can be expressed as:

$$x[n] = \cos(n\omega + \beta_1\pi) + j\sin(n\omega + \beta_2\pi) \quad (5)$$
$$x[n+16] = \cos((n+16)\omega + \beta_3\pi) + j\sin((n+16)\omega + \beta_2\pi)$$

where $\beta_1, \beta_2, \beta_3 \in \{0, 1\}$ and $n \in [10L, 10L + 4)$. In this equation β_1 can either be equal to, or different from β_2 and β_3. Plugging in this relationship into Eq. 5 and taking $arctan$ yields:

$$\angle p[n] = \begin{cases} \pm 16\omega \\ \pm (2n\omega + 16\omega + (2\beta_2 + 1)\pi) \end{cases}$$

Plugging in all possible combinations of parameters and n, the above equation yields a total of 17 discrete values of $\angle p[n] = \pm\frac{\pi}{10}i$, where $i \in \{0, 1, ..., 8\}$.

APPENDIX B
CHANNEL FREQUENCY OFFSET COMPENSATION

For the sake of simplicity, the phase difference of $\angle p[n]$ used throughout the paper assumes channel frequency offset has been compensated. Here we demonstrate how the compensation is computationally achieved. To start, channel frequency offset indicates the difference in the central frequencies of WiFi and ZigBee channels. We note that no ZigBee and WiFi channels have the same central frequencies, even among the ones with overlapping frequencies. For example, while ZigBee channels 11-14 overlap with WiFi channel 1, they deviate from the central frequency of WiFi channel 1 by up to 8 MHz. The offset f_Δ can differ depending on the combinations of WiFi and ZigBee channels. However, interestingly, its impact on $\angle p[n]$ is constant. This is due to the 5MHz channel spacing of ZigBee; The distance of one WiFi channel to any overlapping ZigBee channels are $(3 + 5m)MHz$, $m \in \{-2, -1, 0, 1\}$, where $-2MHz$ is the frequency difference between the WiFi and the closest ZigBee channels – e.g., ZigBee Ch.12 (2.410 GHz) and WiFi Ch.1 (2.412 GHz). By letting $\angle p'[n]$ denote the phase under frequency offset f_Δ (i.e., before compensation), this can be computed by:

$$\angle p'[n] = tan^{-1}(x[n]e^{j2\pi f_\Delta n T_s}x^*[n+16]e^{-j2\pi f_\Delta(n+16)T_s})$$
$$= \angle p[n] - 32\pi f_\Delta T_s$$

where replacing f_Δ with any of the offsets (and plugging in $T_s = 0.05\mu s$) yields the consistent value of $-\frac{24\pi}{5}$. This indicates that any offsets between different channel combinations can commonly be compensated simply by adding $\frac{4\pi}{5}$.

ACKNOWLEDGEMENT

REFERENCES

[1] 802.11 REFERENCE DESIGN FOR WARP V3. https://warpproject.org/trac/wiki/802.11.

[2] REALITY CHECK: 50B IoT DEVICES CONNECTED BY 2020 ? BEYOND THE HYPE AND INTO REALITY. http://www.rcrwireless.com/.

[3] USRP N210. https://www.ettus.com/product/details/UN210-KIT, n.d.

[4] S. S. Adhatarao, M. Arumaithurai, D. Kutscher, and X. Fu. Isi: Integrate sensor networks to internet with icn. *IEEE Internet of Things Journal*, 2017.

[5] F. Adib, S. Kumar, O. Aryan, S. Gollakota, and D. Katabi. Interference alignment by motion. In *MobiCom*, pages 279–290. ACM, 2013.

[6] A. Chakraborty and S. R. Das. Measurement-augmented spectrum databases for white space spectrum. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 67–74. ACM, 2014.

[7] K. Chebrolu and A. Dhekne. Esense: communication through energy sensing. In *MobiCom*, pages 85–96. ACM, 2009.

[8] Z. Chi, Z. Huang, Y. Yao, T. Xie, H. Sun, and T. Zhu. Emf: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous iot devices. In *INFOCOM*, 2017.

[9] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu. B 2 w 2: N-way concurrent communication for iot devices. In *SenSys*, pages 245–258. ACM, 2016.

[10] N. Faruk, K. Ruttik, E. Mutafungwa, and R. Jäntti. Energy savings through self-backhauling for future heterogeneous networks. *Energy*, 115:711–721, 2016.

[11] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 14(2), 2007.

[12] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the rf smog: making 802.11 n robust to cross-technology interference. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 170–181. ACM, 2011.

[13] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy. Crosszig: Combating cross-technology interference in low-power wireless networks. In *IPSN*, pages 1–12. IEEE, 2016.

[14] 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.

[15] W. Jiang, Z. Yin, S. M. Kim, and T. He. Transparent crosstechnology communication over data traffic. In *INFOCOM*, 2017.

[16] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, and T. He. Bluebee: a 10,000x faster cross-technology communication via phy emulation. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2017.

[17] 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.

[18] I. Khan and J. Härri. Can ieee 802.11 p and wi-fi coexist in the 5.9 ghz its band? In *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2017 IEEE 18th International Symposium on*, pages 1–6. IEEE, 2017.

[19] S. M. Kim and T. He. Freebee: Cross-technology communication via free side-channel. In *MobiCom*, pages 317–330. ACM, 2015.

[20] Z. Li and T. He. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2017.

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

[22] A. Misra and K. K. Sarma. Self-organization and optimization in heterogenous networks. In *Interference Mitigation and Energy Management in 5G Heterogeneous Cellular Networks*, pages 246–268. IGI Global, 2017.

[23] A. Rao, O. Schelén, and A. Lindgren. Performance implications for iot over information centric networks. In *Proceedings of the Eleventh ACM Workshop on Challenged Networks*, pages 57–62. ACM, 2016.

[24] S. Rathinakumar, B. Radunovic, and M. K. Marina. Cprecycle: Recycling cyclic prefix for versatile interference mitigation in ofdm based wireless systems. In *CoNEXT*, pages 67–81. ACM, 2016.

[25] T. Salonidis, T. H. Kim, and H. Lundgren. Method of, and apparatus for, controlling a wireless connection in a mimo system using multi-sector directional antennas, June 21 2016. US Patent 9,374,140.

[26] 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 *DYSPAN*, 2012.

[27] R. K. Sheshadri, K. Sundaresan, E. Chai, A. Khojastepour, S. Rangarajan, and D. Koutsonikolas. Blu: Blue-printing interference for robust lte access in unlicensed spectrum. 2017.

[28] F. Song, Z.-Y. Ai, J.-J. Li, G. Pau, M. Collotta, I. You, and H.-K. Zhang. Smart collaborative caching for information-centric iot in fog computing. *Sensors*, 17(11):2512, 2017.

[29] T. Sparber, C. A. Boano, S. S. Kanhere, and K. Römer. Mitigating radio interference in large iot networks through dynamic cca adjustment. *Open Journal of Internet Of Things (OJIOT)*, 3(1):103–113, 2017.

[30] D. H. Staelin. Fast folding algorithm for detection of periodic pulse trains. In *IEEE Proceedings*, volume 57, pages 724–725, 1969.

[31] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[32] X. Wang, X. Lin, Q. Wang, and W. Luan. Mobility increases the connectivity of wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 21(2):440–454, 2013.

[33] Y. Xie, Z. Li, M. Li, and K. Jamieson. Augmenting wide-band 802.11 transmissions via unequal packet bit protection. In *INFOCOM*, pages 1–9. IEEE, 2016.

[34] Z. Yin, W. Jiang, S. M. Kim, and T. He. C-morse: Cross-technology communication with transparent morse coding. In *INFOCOM*, 2017.

[35] Y. Yubo, Y. Panlong, L. Xiangyang, T. Yue, Z. Lan, and Y. Lizhao. Zimo: building cross-technology mimo to harmonize zigbee smog with wifi flash without intervention. In *MobiCom*, pages 465–476. ACM, 2013.

[36] X. Zhang and K. G. Shin. Cooperative carrier signaling: Harmonizing coexisting wpan and wlan devices. *IEEE/ACM ToN*, 2013.

[37] X. Zhang and K. G. Shin. Gap sense: Lightweight coordination of heterogeneous wireless devices. In *INFOCOM*, pages 3094–3101. IEEE, 2013.

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

[39] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma. Zifi: wireless lan discovery via zigbee interference signatures. In *MobiCom*, 2010.

[40] W. Zhou, T. Das, L. Chen, K. Srinivasan, and P. Sinha. Basic: backbone-assisted successive interference cancellation. In *MobiCom*, pages 149–161. ACM, 2016.