BumbleBee: Enabling the Vision of Pervasive ZigBee Backscatter Communication

Zhaoyuan Xu and Wei Gong*
University of Science and Technology of China
xzyjyx@mail.ustc.edu.cn, weigong@ustc.edu.cn

***Abstract*—We present BumbleBee, a novel backscatter system that creates ZigBee transmissions over productive Bluetooth Low Energy (BLE) carriers. In contrast to prior content-aware or non-productive backscatter, BumbleBee overwrites tag information independently on any ambient BLE. The backscattered signal is dominated by the tag information and compliant with commodity ZigBee radios. Since BLE signals are widespread, BumbleBee enables the vision of pervasive ZigBee backscatter communication.**

We prototype BumbleBee using commodity BLE transmitters, an off-the-shelf FPGA, and commodity ZigBee receivers. Through extensive experiments and field studies, we show that BumbleBee works universally with ambient BLE and commodity receivers. Further, when the signal strength is -80 dBm, BumbleBee has a throughput of 218 kbps and 204 kbps in the line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios, respectively. The throughput improvement is up to 3x compared with the advanced non-productive backscatter system, Interscatter [1], and 32x over the content-aware backscatter system, FreeRider [2]. The bit error ratio (BER) is below 1% when the tag-to-receiver distance is 20 meters. As the first ambient ZigBee backscatter system that works universally with commodity transceivers, we believe BumbleBee takes a crucial step towards pervasive ZigBee backscatter communication.

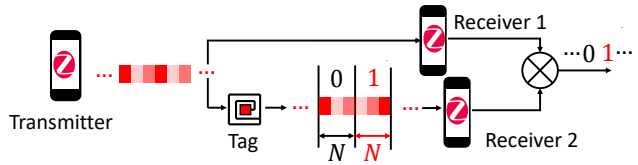
***Index Terms*—System; IoT; Backscatter; ZigBee; BLE**

I. Introduction

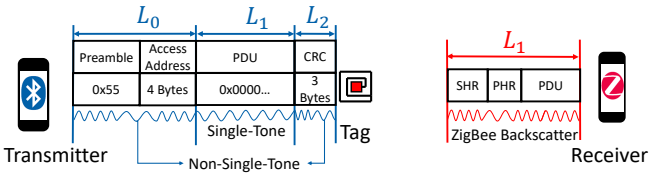
In recent years, ambient backscatter has attracted a lot of interest as it is promising to enable ultra-low-power communications for billions of Internet-of-Things (IoT) devices [1] [2] [3] [4] [5] [6] [7]. Two appealing visions distinguish it from conventional RFID communications [8] [9] [10] [11]. Firstly, it intends to use uncontrolled ambient signals for wireless carriers, removing the dependence on dedicated carrier generation. Secondly, it empowers commodity radios for signal demodulation, expanding backscatter receivers from bulky RFID readers to pervasive IoT that support general-purpose wireless protocols, e.g., ZigBee [1] [2] [7], Wi-Fi [6] [12], BLE [4] [13], etc.

The independence of excitors and receivers makes ambient backscatter an excellent candidate for pervasive communications. But it also adds challenges to the system design since ambient signals are out-of-control. Backscatter is the time-domain product of ambient signals and tag-data modulation, in which receivers work to recover tag data. Ambient signals introduce uncontrolled content (e.g., amplitude, phase, frequency, etc.) to the product and disturb the tag-data demodulation. To solve the problem, advanced backscatter systems restrict either excitors or receivers [1] [2] [14] [15] [16] [17] [18] [19].

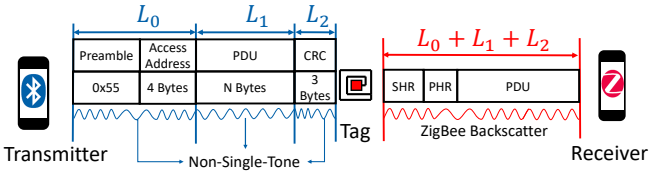
*Corresponding author: Wei Gong



(a) FreeRider. It requires additional ZigBee radios for signal reception. Further, N symbols (N=8 in practice.) are used to modulate a single bit. Since one ZigBee symbol contains four bits for active ZigBee transmission, the throughput of FreeRider is reduced over 32x times.



(b) Interscatter. Specific BLE transmitter provides a single-tone carrier through reversed data whitening. The tag backscatters BLE single-tone to generate ZigBee signals. The maximum carrier utilization is $\frac{L_1}{(L_0+L_1+L_2)}$.



(c) BumbleBee. The tag backscatters productive BLE to generate ZigBee signals. Its carrier utilization can reach $\frac{(L_0+L_1+L_2)}{(L_0+L_1+L_2)} = 1$.

Fig. 1. System overview.

The first class of ambient backscatter exploits content-aware backscatter [2] [16] [18] [19], which employs additional receivers to eliminate ambient uncertainty. In Fig. 1 (a), FreeRider [2] deploys two receivers for the reception of backscattered and corresponding ambient symbols. The tag data is demodulated as '1' when they are different and vice versa to '0'. The key to system reliability is redundant coding. It takes multi-symbols (eight for ZigBee) to encode a single bit, and thus greatly decreases system throughput. Other systems, employing single-tone excitors, exploit non-productive backscatter [1] [14] [15] [17]. The ambient content is minimized (i.e., constant amplitude, phase, and frequency)

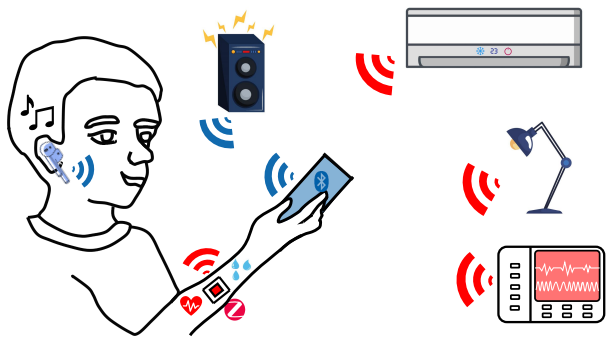


Fig. 2. Application snapshot. BumbleBee enables pervasive ZigBee transmissions with widely spread BLE carriers. The BLE connection that is streaming music can be used to create ZigBee transmissions.

and does little interference on the backscattered signal. As shown in Fig. 1 (b), Interscatter [1] takes commodity BLE radios to generate single-tone carriers. The tag skips the uncontrollable packet field (L_0 and L_2, which are generally non-single-tone.) and backscatters the single-tone filed (L_1) to generate ZigBee signals. However, despite the dependency of specific single-tone excitors, its carrier utilization is limited to $\frac{L_1}{L_0+L_1+L_2}$, and throughput is correspondingly constrained by the single-tone length.

In a word, those works have the following drawbacks: 1) they suffer from either single-tone excitors, which are dedicated to constant content transmission and fail to communicate with other nodes, or extra receivers with increasing deployment cost. 2) their throughput is limited due to the essential redundant coding or the low carrier utilization. Thus, we ask a simple but difficult question: is it possible to build a productive ZigBee backscatter system with ambient BLE excitations? The positive answer envisions our tags to reuse widespread wireless carriers and only one commodity radio is adequate to demodulate tag data. However, such a system is difficult to design since uncontrolled ambient content delivers disruptive interference on the backscattered product.

We present BumbleBee, a novel backscatter system that creates ZigBee transmissions over productive BLE carriers. As shown in Fig. 1 (c), BumbleBee reuses arbitrary BLE to backscatter ZigBee and only requires one commodity ZigBee receiver. Its carrier utilization can reach $\frac{L_0+L_1+L_2}{L_0+L_1+L_2} = 100\%$ and throughput is correspondingly improved ($L_1 \rightarrow L_0+L_1+L_2$). Further, it also allows us to benefit from the increasing usage of BLE and ZigBee radios in our life. As shown in Fig. 2, connections of smartphones, headsets, and smart speakers are ready for BumbleBee excitors. People will be pleased with BumbleBee because they can get tag data while listening, calling, or transmitting files. The implementation of BumbleBee is challenging since BLE and ZigBee have different physical layer specifications. Specifically, BLE reaches a bitrate cap of 1 Mbps and adopts Gaussian Frequency Shift Keying (GFSK) modulation. ZigBee has a bitrate of 250 kbps and takes both Offset Quadrature Phase Shift Keying (OQPSK) and Direct-Sequence Spread Spectrum (DSSS) for content modulation. The transformation from productive BLE to ZigBee has no theoretical or experimental support. Uncontrolled GFSK makes the carrier phase a time-varying component while conventional ZigBee backscatter tags use phase modulation to piggyback content. A conflict is created since commodity ZigBee radios take consecutive sampling phases for signal demodulation. Tag data can be eliminated by carrier interference.

BumbleBee takes a simple but crucial design toward pervasive ZigBee backscatter communication. At a high level, it introduces dominant tag data on the backscattered signal. Ambient BLE content is overwritten and commodity ZigBee radios are robust to recover tag data. Specifically, in this paper, we make the following contributions.

- We provide an insightful observation that productive BLE, which is pervasive in our lives, is a qualified RF carrier for ambient ZigBee backscatter. Its phase shift within each ZigBee chip unit is concentrated within [-1, 1] and left rich space for tag data modulation.
- We propose BumbleBee, a novel ambient backscatter system that overwrites dominant tag data on ambient BLE. It shows that backscattered products, suffering from ambient interference, can also be recovered by receiver robustness. The backscattered signal works compliantly with commodity radios and only one commodity radio is adequate to recover tag data.
- We build a prototype of BumbleBee and validate its effectiveness through extensive experiments. Specifically, when the signal strength is -80 dBm, BumbleBee has a throughput of 218 kbps and 204 kbps in the LOS and NLOS scenarios, respectively. The bit error ratio (BER) is below 1% when the tag-to-receiver distance is 20 meters.

II. Preliminaries

A. Commodity BLE Transmitter

The architecture of commodity BLE transmitters is shown in Fig. 3, which supports a bitrate cap of 1 Mbps [20]. A BFSK module takes the bitstream as input. It assigns bit '1' to a frequency deviation of +250 kHz and bit '0' to -250 kHz. A Gaussian Filter (GF), pulse shaping an input frequency into a specific waveform, is realized with an oversampling of 13 [21]. Its impulse response is shown in Eq. (1), where K_{BT} is calculated as the channel bandwidth, N is the symbol rate (For BLE, one bit represents one symbol), and erf is the error function. The complete output of the Gaussian pulse overlaps consecutive L BLE symbols. Specifically, when L is set to 3, the output ($\Delta f(n)$) of GF is shown in Eq. (2). g_0, g_1, and g_2 are the previous, instantaneous, and successive pulse responses, respectively.

$$\begin{aligned} g(n) = \quad & \frac{1}{4N}[erf(\pi K_{BT}\sqrt{\frac{2}{In2}}(\frac{n}{N}+\frac{1}{2})) \\ & -erf(\pi K_{BT}\sqrt{\frac{2}{In2}}(\frac{n}{N}-\frac{1}{2}))] \\ & n \in [-\frac{L*N}{2}, +\frac{L*N}{2}] \end{aligned} \tag{1}$$

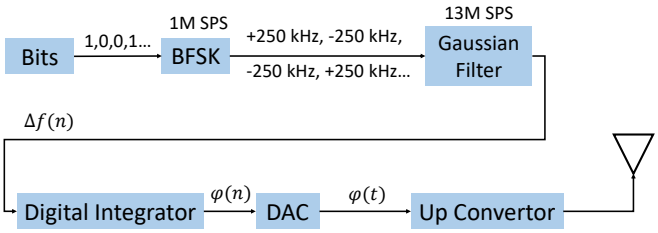


Fig. 3. Architecture of commodity BLE transmitter.

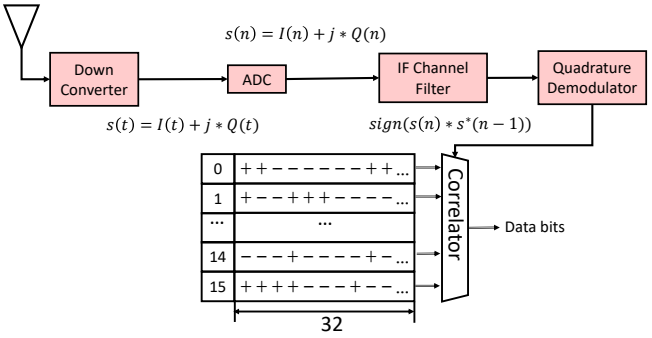


Fig. 4. Architecture of commodity ZigBee receiver.

$$\Delta f(n) = g_0(n + \frac{N}{3}) + g_1(n) + g_2(n - \frac{N}{3}) \quad (2)$$

A digital integrator works to integrate the GF output ($\Delta f(n)$) into a sequence of consecutive phase shift: $\phi(n) = \phi(n-1) + 2\pi\Delta f(n) * \Delta t$. A digital-to-analog-convertor (DAC) is used to translate digital samples into analog $\phi(t)$. Finally, an up-converter functions to translate the baseband signal into RF signals and transmits it through the antenna.

B. Commodity ZigBee Receiver

The physical layer of ZigBee is IEEE 802.15.4 [22]. It transmits packets at a bitstream of 250 kbps. Every four bits are spread into a pseudo-random noise (PN) chip sequence, which is known as the direct-sequence spread spectrum (DSSS). Every chip unit is T_c ($0.5\mu s$), equivalent to a transmission rate of 2 M chip/s. ZigBee leverages Quadrature Phase Shift Keying (OQPSK) to modulate chip sequence and introduces a phase shift of $\pm\frac{\pi}{2}$ every T_c.

The architecture of the commodity ZigBee receiver is shown in Fig. 4. The RF signal is first conducted through an antenna and down-converted to the baseband. An analog-to-digital converter (ADC) transforms the analog baseband $s(t)$ into the digital domain $s(n)$. An Intermediate Frequency (IF) channel filter takes the digital samples as input and functions to eliminate out-of-band noise. A quadrature demodulator detects the filtered phase shift sequence every T_c. It is not sensitive to concrete values, but only concerns the sign of consecutive phase shift. Specifically, the phase shift sequence is calculated as:

$$sequence(n) = [sign(s(n) * s^*(n-1)), sign(s(n-1) * s^*(n-2)), ..., sign(s(n-30) * s^*(n-31)), sign(s(n-31) * s^*(n-32))] \quad (3)$$

$s^*(n)$ denotes the conjugate of $s(n)$. Further, the sequence is also correlated with the predefined symbol sequence. The closest symbol, which has the minimum Hamming distance with the input sequence, is despread into a bitstream.

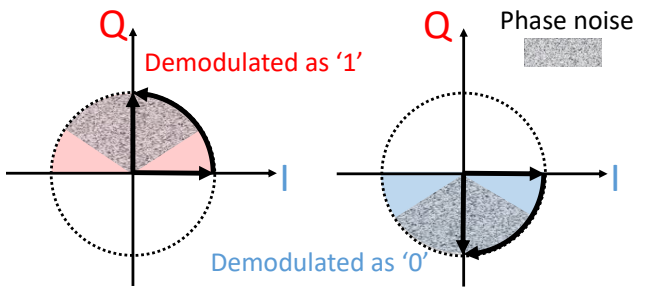
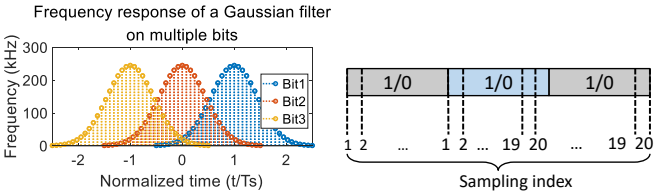


Fig. 5. Overwrite modulation is robust to ambient BLE. Our tag overwrites a phase shift of $\pm\frac{\pi}{2}$ to piggyback '0' or '1'. Productive BLE introduces phase noise on the backscattered signal. However, it will not affect the tag-data demodulation since the tag data is dominant over ambient BLE.



(a) The output of Gaussian Filter is overlapped by consecutive BLE bits.

(b) The sampling rate is 20 MSPS. The phase shift is calculated every T_c (0.5 μs).

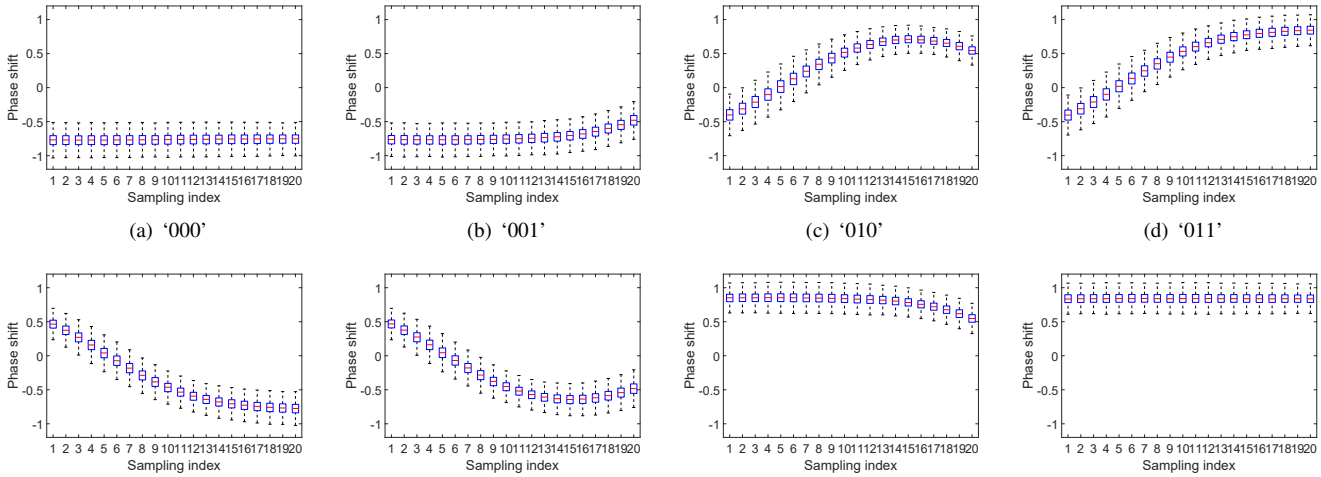
Fig. 6. Sampling deployments for BLE observation.

III. Design

We use backscatter to transform productive BLE transmissions into ZigBee signals. In this section, we first present the basic idea of BumbleBee, which overwrites tag data independently on ambient BLE. Next, we demonstrate the regulations of BLE phase shift within each ZigBee chip unit. Finally, we outline the design and implementation of BumbleBee.

A. Basic Idea

Conventional non-productive backscatter systems [1] [12] [23] insist on providing content-invariant (i.e., constant amplitude, phase, and frequency) carriers for tag-data modulation. The excitor has to give up data transmission in order to provide the non-interference RF carriers. Differently, we look at the problem from a novel perspective: if we can modulate dominant tag data over ambient carriers, we do not need to



(a) '000' (b) '001' (c) '010' (d) '011'

(e) '100' (f) '101' (g) '110' (h) '111'

Fig. 7. Phase shift of representative BLE patterns.

constrain the carrier content, and thus it is promising to build a productive backscatter system.

BumbleBee follows the idea and exploits the characteristics of both the BLE transmitter and ZigBee receiver. On the one hand, ambient BLE provides an opportunity for ZigBee generation since its phase shift within the ZigBee chip unit is negligible ($\in [-1, 1]$). Detailed evaluations are shown in the following subsections. On the other hand, ZigBee receivers use a quadrature demodulator for phase content demodulation. The demodulation rules are shown below:

$$decision = \begin{cases} 1, & s(n) * s^*(n-1) \in [0, \pi] \\ 0, & s(n) * s^*(n-1) \in [-\pi, 0] \end{cases} \quad (4)$$

As shown in Fig. 5, productive BLE introduces phase noise on the backscattered signal, but the native phase content is still recoverable. The backscattered signal is demodulated to '1' when the tag data is $+\frac{\pi}{2}$ and '0' when it comes to $-\frac{\pi}{2}$.

Along this line, we will first observe the phase shift of BLE within the ZigBee chip unit. After that, we will show the BumbleBee tag design and demonstrate its effectiveness for ZigBee generation.

B. BLE Phase Shift

To completely obtain an overview of the BLE phase shift, an intuitive approach is to traverse all BLE packets and obtain their distribution. Such an experiment is easy to design but difficult to deploy in practice. Specifically, a BLE advertising packet supports a payload length of 37 bytes. This corresponds to over $2^{(37*8)} \approx 10^{89}$ possibilities and is greatly beyond our estimation. To reduce the complexity, we are wondering if we can determine the BLE phase shift by some representative BLE patterns.

BLE pattern. An inspiration comes from the commodity BLE transmitter. As shown in Fig. 6 (a), the output of the Gaussian Filter is overlapped by consecutive BLE impulse response. One BLE bit cannot independently determine the corresponding phase shift, which is also overlapped by neighbor BLE pulses. [21] demonstrates that one BLE bit can overlap consecutive 3 Gaussian pulses. The previous and successive Gaussian pulses can affect the instantaneous pulses concurrently. Thus, BLE packets are split into eight different patterns (i.e., '000', '001', '010', '011', '100', '101', '110', '111'). We are able to identify BLE phase shifts by observing these representative patterns. Specifically, the phase shift in the second bit is representative to observe since it can only be overlapped by the first and third BLE bits. Both the phase shift in the first and third bit can be affected by unknown pulses.

In the following evaluations, we use TI CC2640R2F [24] as the BLE transmitter and take HackRF [25] to capture signals over the air. BLE packets are randomly transmitted and the sampling rate of the receiver is set to 20 M samples per second (SPS). Our experiment setup is shown in Fig. 6 (b). The phase shift over the second bit is calculated every ZigBee chip unit ($T_c = 0.5\mu s$). Over 840000 sampling points are counted and each pattern is counted over 5000 times at random packets. Further, the phase shift of overall BLE packets is also counted to picture the distribution of random BLE.

Phase shift distribution. We measure the phase shift distribution over the second bit for different BLE patterns. As shown in Fig. 7, pattern '000' and '111' concentrate around ± 0.7 ($\approx \pm\frac{\pi}{4}$). Other patterns, including '001', '010', '011', '100', '101', and '110' attain a tradeoff between '111' and '000'. Specifically, their median concentration will either fall below that of '000', nor exceed '111' and the specific distribution is determined by neighbor BLE pulses. Since every BLE packet consists of representative patterns and all of them are concentrated within [-1, 1], it is easy to identify that the phase shift of ambient BLE is also concentrated within [-1, 1]. To confirm the conclusion, we also count the phase shift of random BLE in Fig. 8. Most of them are distributed between [-1, 1].

The observation demonstrates the following facts: 1) Differ-

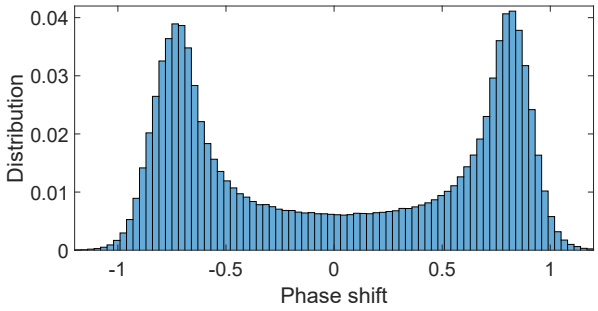


Fig. 8. Distribution of ambient BLE.

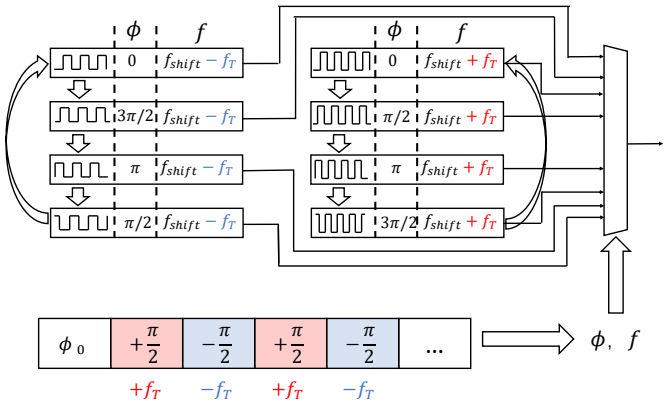


Fig. 9. The clock generation and data modulation of BumbleBee.

ent patterns cover different ranges. The sign of phase shift over the second BLE bit is subject to neighbor BLE pulses. 2) Fig. 8 shows that the phase shift of random BLE is symmetrically distributed within [-1, 1].

C. BumbleBee Design

Dominant phase shift. The value of the dominant phase shift is a crucial design for BumbleBee implementation. Small phase shifts make it difficult to overwrite ambient content, while large ones may cause out-of-bounds demodulation at the receiver. This question seems hard to answer, but we can figure out the optimum product from the characteristics of the system transceiver. As aforementioned, the phase shift of ambient BLE is symmetrically distributed around 0 and most of them are concentrated within [-1, 1]. It is much smaller than that of the ZigBee receiver ($[-\pi, 0]$ or $[0, \pi]$) in Eq. (4), which provides an opportunity for our tag to modulate a dominant phase content. $\pm\frac{\pi}{2}$ are in the center of ZigBee boundaries ($\frac{(\pi+0)}{2} = +\frac{\pi}{2}, \frac{(-\pi+0)}{2} = -\frac{\pi}{2}$) and left a rich space for data recovery:

$$0 \leq +\frac{\pi}{2} \pm 1 \leq +\pi \tag{5}$$

$$-\pi \leq -\frac{\pi}{2} \pm 1 \leq 0 \tag{6}$$

Specifically, when it comes to '1', BumbleBee modulates a positive phase shift of $+\frac{\pi}{2}$ on the backscattered signal. And when it comes to '0', a negative phase shift of $-\frac{\pi}{2}$ is induced. Next, a question arises: how to enable backscatter signals with a phase shift of $\pm\frac{\pi}{2}$ every T_c?

Intuitive design. The excitation is shown in Eq. (7). A_c, f_c, f_{BLE} and ϕ_{BLE} are the excitation amplitude, central frequency, frequency deviation and phase content, respectively. An intuitive idea is to schedule four square waves with different phases ($f_T = 0$, $\phi_T \in \{0, +\frac{\pi}{2}, \pi, -\frac{\pi}{2}\}$) for the backscatter signal generation. Since the signal strength is generally much higher than that of backscatter, all of the square waves have a frequency deviation of f_{shift}, which shifts the backscatter signal from the excitation channel to another. As shown in Eq. (8), square waves can be written with a combination of multiple sinusoidal signals. A_T, $(f_{shift}+f_T)$, and ϕ_T denotes the tag amplitude, frequency, and phase states, respectively. Only the first harmonic is the dominant and desired term.

$$C(t) = A_c e^{j(2\pi(f_c+f_{BLE})t+\phi_{BLE})} \tag{7}$$

$$\begin{aligned} T(t) &= A_T((cos(2\pi(f_{shift}+f_T)t+\phi_T) \\ &+ \sum_{n=3,5,7,\ldots}^{\infty} \frac{1}{n} cos(2\pi n f_{shift} t + \phi_T) \\ &+ jsin(2\pi(f_{shift}+f_T)t+\phi_T) \\ &+ \sum_{n=3,5,7,\ldots}^{\infty} \frac{1}{n} jsin(2\pi n f_{shift} t + \phi_T))) \\ &= A_T e^{j(2\pi(f_{shift}+f_T)t+\phi_T)} \end{aligned} \tag{8}$$

$$\begin{aligned} B(t) &= C(t)T(t) \\ &= A_c e^{j(2\pi(f_c+f_{BLE})t+\phi_{BLE})} A_T e^{j(2\pi(f_{shift}+f_T)t+\phi_T)} \\ &= A_c A_T e^{j(2\pi(f_c+f_{shift}+f_{BLE}+f_T)t+(\phi_{BLE}+\phi_T))} \\ \phi_T &\in \{0, \frac{\pi}{2}, \pi, -\frac{\pi}{2}\} \end{aligned} \tag{9}$$

The phase states are shifted instantaneously every T_c. When it comes to modulate '1', the square wave with a positive phase shift ($\phi_0 \rightarrow (\phi_0 + \frac{\pi}{2})$) is scheduled to control the RF switch. Correspondingly, the square wave with a negative phase shift ($\phi_0 \rightarrow (\phi_0 - \frac{\pi}{2})$) is scheduled to modulate '0'. The backscattered signal is shown in Eq. (9). For simplicity, other (e.g., the second, third, ... etc) harmonics are omitted.

Frequency phase shift modulation. In this paper, the instantaneous phase shift (e.g., $0 \rightarrow +\frac{\pi}{2}$, $+\frac{\pi}{2} \rightarrow 0$) is denoted as IPS modulation. It is easy to follow whereas attaining poor spectrum efficiency. For example, the Fourier series of the instantaneous phase shift sequence ($0 \rightarrow +\frac{\pi}{2} \rightarrow +\pi \rightarrow -\frac{\pi}{2}$...) has infinite harmonics so that increasing the spectrum occupation. In Fig. 13 (b), an IPS backscatter system, Interscatter [1], has a spectrum bandwidth of 5.7 MHz. It is much higher than that of commodity radios (1.5 MHz). Further, since the excitation bandwidth of BumbleBee is much greater than that of Interscatter, it is a dilemma for IPS-based tags to work in the crowded spectrum. Differently, we adopt Frequency-Phase Shift (FPS) modulation [7] to reduce the spectrum occupation. It uses an additional frequency shift $\pm f_T = \pm\frac{\frac{\pi}{2}}{2\pi T_c} = \pm\frac{1}{4T_c}$

to accumulate $\pm\frac{\pi}{2}$ every T_c. Specifically, the sequence ($0 \rightarrow +\frac{\pi}{2} \rightarrow +\pi \rightarrow -\frac{\pi}{2}$...) can be completed by only one Fourier series $e^{j2\pi f_T t}$, which greatly improves the spectrum efficiency. The design is nice in theory, and listed below in detail.

State machine As shown in Fig. 9, eight square waves are scheduled to complete the FPS modulation. The baseband signal consists of a phase shift sequence (e.g., 0, $+\frac{\pi}{2}$, $-\frac{\pi}{2}$, $+\frac{\pi}{2}$, $-\frac{\pi}{2}$,...) with an initial phase ϕ_0. The frequency deviation ($f_{shift} \pm f_T$) is correspondingly determined by the sign ($\pm$) of phase shift. Both of the accumulated phase shift and the frequency deviation are directed for the selection of the multiplexer. It is notable that the states of square waves are determined by the relative phase to the reference clock ($f = f_{shift}, \phi = 0$) so that cyclically transform. For example, at the beginning, one square wave generated with ($\phi = 0, f = f_{shift} - f_T$) is selected when the selection of multiplexer is ($\phi = 0, f = -$). After T_c, the signal is selected only when the multiplexer selection is ($\phi = -\frac{\pi}{2}, f = -$). The clock transforms its state cyclically since its relative phase to a reference clock is transformed. The specific rules of phase state transformation are shown in Fig. 9.

D. ZigBee Packet Assembly

In this section, we show how to construct ZigBee packets using ambient BLE. We take BLE advertising packets for example and show their difference with non-productive backscatter. The specific structure of the BLE advertising packet is shown in Fig. 10. It consists of a preamble, access address, packet header, advertising address, payload, and CRC. Interscatter uses an envelope detector for BLE signal detection and finds the start of single-tone by skipping a guard interval. A complete ZigBee packet consists of a preamble, synchronization header (SHR), payload, and CRC. Limited by the length of BLE single-tone (37*8=296 microseconds), it can only construct a ZigBee packet with a payload length of 1 byte. BumbleBee uses overwrite modulation and takes productive BLE for RF carriers. In particular, it uses an envelope detector for BLE detection and then constructs the preamble, SHR, PHR, and payload in sequence. The maximum payload length supported by advertising BLE is calculated as 3 bytes. Theoretically, it can reach 100% utilization for BLE carriers, which greatly expands the usage of BumbleBee.

E. Demodulation of Excitation Instructions

We adopt ON-OFF keying to demodulate data bits transmitted by the excitation source, which follows the design shown in [12]. Specifically, a bit '1' is decoded for a high amplitude at a length of T_1 while bit '0' is decoded for a low amplitude at a length of T_0. We design a packet structure to decode ambient signals, which consists of a synchronization header, a PHY header, and a PHY service data unit (following the structure in IEEE 802.15.4). Only the packets whose header complies with the synchronization header design can be further demodulated.

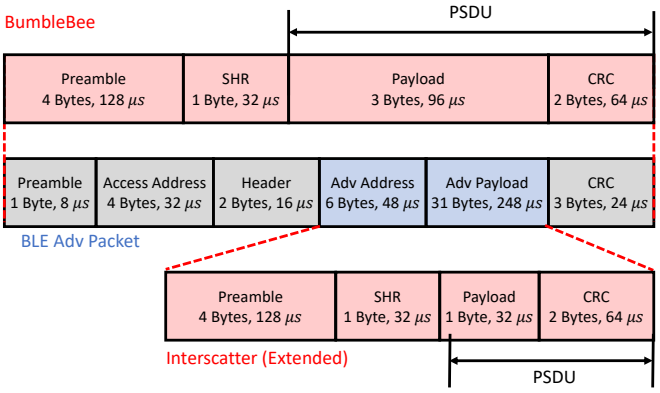


Fig. 10. Structure comparison of BumbleBee and Interscatter taking BLE advertising packets.

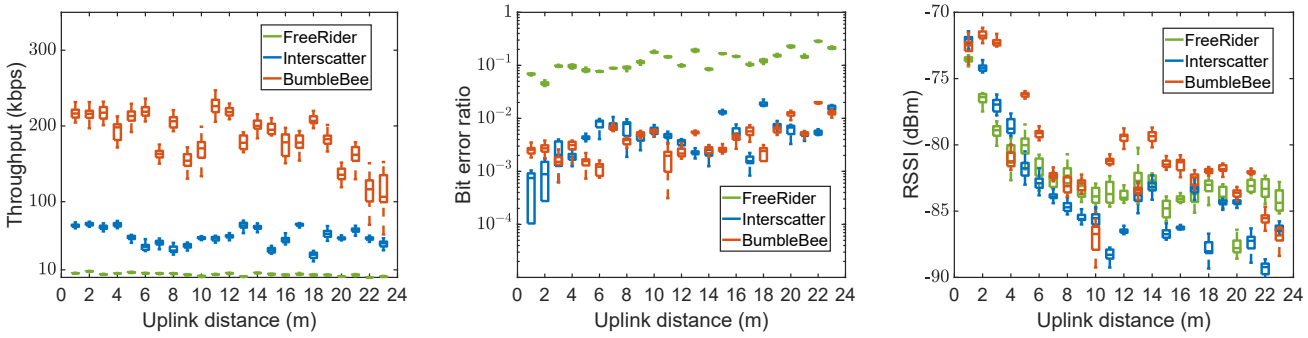
IV. Implementation

BLE transmitters and ZigBee receivers We use commodity radios TI CC2640R2F [24] for BLE transmission and CC2530 [26] for ZigBee reception. We also replace BLE transmitter with CC1352 [27], CC2540 [28], and CC2650 [29], and ZigBee receiver with CC2650 [29] to evaluate the universality of BumbleBee. The transmitter is connected to a power amplifier, whose up gain is 17$\pm$ 3 dB, and the transmission rate is 23 packets/s. For simplicity, the transmission channel is set to BLE channel 3 (2410 MHz) and the reception channel is ZigBee channel 14 (2420 MHz). Interscatter [1] and FreeRider [2] are prototyped using the same hardware as BumbleBee. Differently, the excitation of Interscatter is replaced with BLE single-tone. The excitation of FreeRider is ATMEGA256RF2 [30]. The transmission length is 60 bytes and the redundant coding is set to eight, which takes eight ZigBee symbols to modulate a single bit.

Backscatter tag The prototype of BumbleBee consists of an RF front-end circuit and an FPGA. The RF front end includes an envelope detector, a comparator, and an RF switch. The envelope detector is AD8313 [31], whose output is connected to a comparator. The comparator sets a threshold to eliminate ambient noise and decode downlink instructions. The overwrite modulation works as soon as the comparator output is true. The RF switch is ADG902 [32] and has different impedance loads. It is connected to an FPGA (XILINX ZYNQ 7000) for backscatter signals generation. The frequency shift (f_{shift}) is set to 10 MHz and frequency deviation (f_d) is set to 0.5 MHz. Eight square waves ($f \in \{f_{shift} + f_d, f_{shift} - f_d\}, \phi \in \{0, \frac{\pi}{2}, \pi, -\frac{\pi}{2}\}$) are reassembled for the generation of BumbleBee.

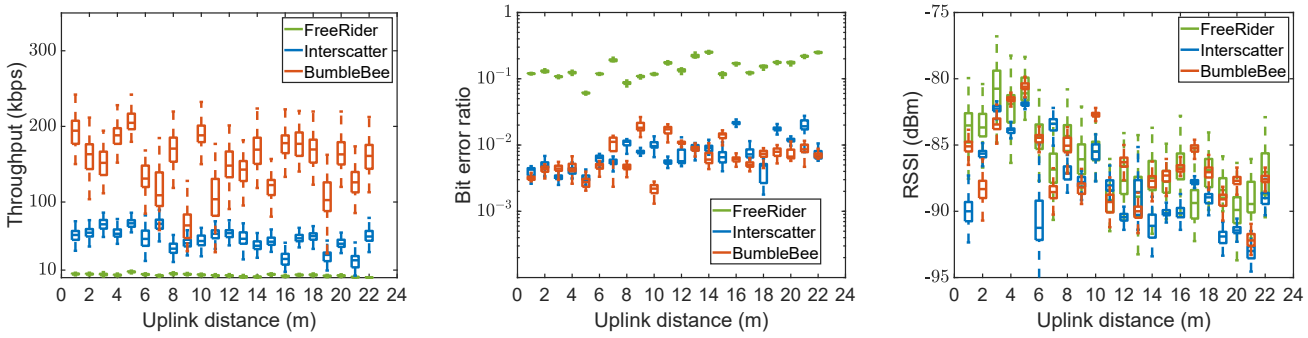
V. Evaluation

We first evaluate BumbleBee's end-to-end performance and then show its universality with commodity radios. Next, we investigate the spectrum efficiency and co-existence with am-



(a) Throughput (b) BER (c) RSSI

Fig. 11. Backscatter throughput, BER, and RSSI in the LOS scenarios.



(a) Throughput (b) BER (c) RSSI

Fig. 12. Backscatter throughput, BER, and RSSI in the NLOS scenarios.

bient BLE. Finally, we present the distribution of throughput and RSSI in the real world.

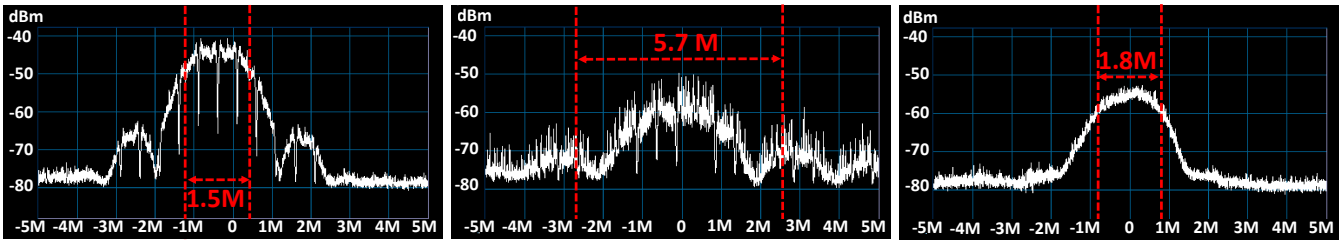
A. End-to-End Performance

Line-of-Sight (LOS). Fig. 11 evaluates the system throughput, BER, and RSSI in the LOS scenarios. It is worth noting that the throughput of BumbleBee is calculated as 3x over Interscatter [1] and 32x over FreeRider [2]. which contributes to the overwrite modulation without waiting for ambient single-tone and redundant coding. Fig. 11(a) shows that when the uplink distance is 6 meters, BumbleBee has a throughput of 218 kbps and its corresponding RSSI is -80 dBm. In comparison, the throughput of Interscatter and FreeRider is 40 kbps and 4.7 kbps. The maximum throughput of BumbleBee, Interscatter, and FreeRider is 226 kbps, 68 kbps, and 8 kbps. And the minimum throughput is 106 kbps, 32 kbps, and 0.5 kbps. BumbleBee also achieves a throughput of 106 kbps when the uplink distance is 23 meters and RSSI is -86 dBm. Further, Fig. 11(b) shows that BumbleBee has a comparable BER to Interscatter (non-productive backscatter) in the LOS scenarios. Its BER is below 1% when the uplink distance is 20 meters. FreeRider has a BER of over 4%. The minimum BER of BumbleBee, Interscatter, and FreeRider is 0.1%, 0.075%, and 4.5 %. And the maximum BER is 1.9%, 1.7%, and 29%, respectively. Fig. 11(c) shows that the signal strength of various systems is close to each other and decreases gradually with the uplink distance. The RSSI varies from -71 dBm to -90 dBm.

Non-Line-of-Sight (NLOS). Fig. 12 evaluates the system throughput, BER, and RSSI in the NLOS scenarios. In Fig. 12(a), BumbleBee has a throughput of 204 kbps and its corresponding RSSI is -80 dBm. The throughput of Interscatter and FreeRider is 71 kbps and 7 kbps, which varies widely in different positions (compared to the LOS scenarios) due to the blocking of objects. BumbleBee achieves a throughput of 160 kbps at -87 dBm when the uplink distance is 22 meters. The maximum throughput of BumbleBee, Interscatter, and FreeRider is 204 kbps, 71 kbps, and 7 kbps, whereas their minimum throughput is 69 kbps, 22 kbps, and 0.3 kbps. Fig. 12(b) shows that the BER of BumbleBee and Interscatter are comparable to each other in the NLOS scenarios. The BER of BumbleBee does not exceed 1% when the uplink distance is 6 meters. In comparison, the BER of FreeRider is above 6% in most of the deployments. The minimum BER of BumbleBee, Interscatter, and FreeRider is 0.2%, 0.3%, and 6%. And the maximum BER is 1.8%, 2%, and 25%, respectively. In Fig. 12(c), the system signal strength of is close to each other, which varies from -80 dBm to -93 dBm.

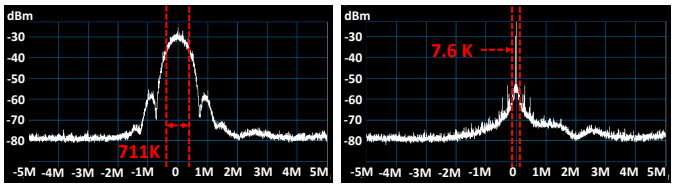
B. Spectrum Efficiency

Since BumbleBee takes ambient BLE signals for RF carriers, we want to explore whether it consumes additional band-



(a) Commodity ZigBee (b) Interscatter (c) BumbleBee

Fig. 13. Spectrum comparison of commodity ZigBee, Interscatter, and BumbleBee.



(a) Spectrum of commodity BLE. (b) Spectrum of single-tone.

Fig. 14. The spectrum of different carriers.

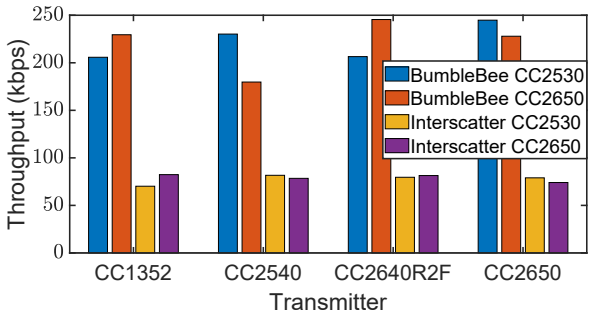


Fig. 15. Throughput comparison of commodity transceivers.

width compared to Interscatter. We first show the bandwidth of the two system carriers, followed by the spectral efficiency of the different systems, which is used to illustrate the frequency domain performance of BumbleBee.

Carrier generation. A PXIe-5663 RF Vector Signal Analyzer is used to measure the occupied bandwidth containing over 90% RF energy. A commodity BLE radio (CC1352) transmits BLE packets and single-tone signals continuously. Fig. 14 shows the bandwidth of different carriers. It demonstrates that a single-tone carrier has a bandwidth of 7.6 kHz while commodity BLE occupies a bandwidth of 711 kHz. In terms of carrier bandwidth, BLE packets occupy a bandwidth that is seven times the rate of a single-frequency continuous wave. We are curious whether using such a carrier for backscatter communication will result in additional bandwidth.

Spectrum efficiency. We evaluate the bandwidth of different modulations and commodity ZigBee. A PXIe-5663 RF Vector Signal Analyzer is used to measure the occupied bandwidth containing over 90% RF energy. A commodity ZigBee radio (CC1352) transmits ZigBee packets continuously. Our tag takes single-tone carriers to modulate Interscatter and productive BLE carriers to modulate BumbleBee. Fig. 13 shows their occupied bandwidth. A commodity ZigBee radio has a bandwidth of 1.5 MHz. Interscatter has a bandwidth of 5.7 MHz, which is 3.8x greater than the active radio. BumbleBee occupies a bandwidth of 1.8 MHz. It is much smaller than that of Interscatter and comparable to the commodity ZigBee. The above results show that although BumbleBee occupies a higher bandwidth, it is still much lower than Interscatter. We believe that it is better than Interscatter in the frequency domain.

C. Universality of BumbleBee

To demonstrate the system universality with commodity transceivers, BumbleBee is evaluated with multiple BLE transmitters (CC1352 [27], CC2540 [28], CC2640R2F [24], CC2650 [29]) and various ZigBee receivers (CC2530 [26], CC2650 [29]). In Fig. 15, the throughput of BumbleBee and Interscatter is evaluated with different pairs of commodity radios. The minimum throughput of BumbleBee is generally over 179 kbps. In comparison, the maximum throughput of Interscatter is below 82 kbps. The performance of BumbleBee varies with BLE transmitters. For example, when we use CC2530 as the receiver, the throughput of BumbleBee (CC2530 and CC2650 receiver) and Interscatter (CC2530 and CC2650 receiver) achieves 205 kbps, 230 kbps, 206 kbps, and 244 kbps, respectively. In addition, the performance also varies with ZigBee receivers. Interscatter has a limited range compared to BumbleBee. For example, when we use CC2540 as the transmitter, Interscatter has a maximum throughput gap of 12 kbps, whereas BumbleBee varies from 230 kbps to 179 kbps with a range of 51 kbps.

D. Co-existence with Ambient BLE

In this section, we explore whether BumbleBee causes interference with BLE transmissions. We used different BLE senders and receivers and tested their throughput rates in both tagged and untagged conditions. As shown in Fig. 16, the effect of tags on raw BLE transmissions is limited under the same conditions of the sender and receiver. For example, when we use CC2540 as the sender and CC1352 as the receiver, the

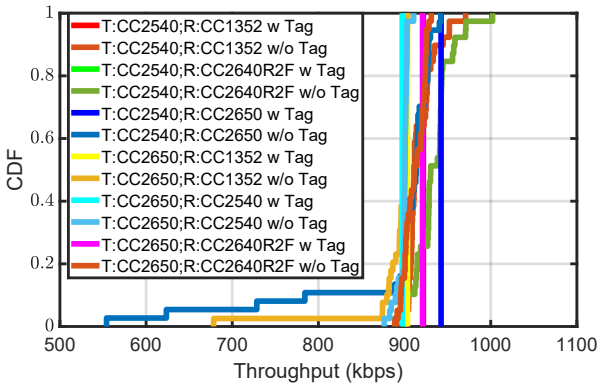


Fig. 16. BumbleBee works co-existing with commodity BLE transmitters.

data transmission with and without tags is around 900 kbps, which proves that BumbleBee can coexist well with ambient BLE.

E. Distribution of Signal Strength and Throughput

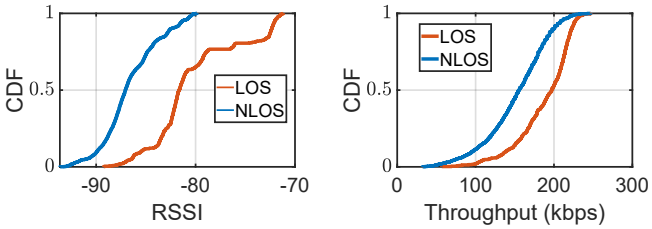
Our tag takes a single-tone carrier for signal excitation and adopts two modulation technologies (i.e., IPS and FPS) for the backscatter packet generation. The tag is placed at different locations and its distance from the excitation source is gradually increased from 1 meter up to 20 meters. Fig. 17 shows their signal strength distribution, respectively. It demonstrates that the backscatter signal strength can be over -60 dBm for practice. Half of the signal strength distribution can be over -70 dBm.

VI. Related work

Various excitation sources, modulation technologies, and synchronization achievement prosper backscatter communications. Researchers design different communication systems for application scenarios with different requirements. The related works are discussed below.

Excitation source Backscatter tag leverages ambient signal ([14], [33], [4], [34], [35], [36], [37], [38]) for excitation carrier to avoid power-consuming RF components. Various RF sources prosper backscatter communication. mmTag [35] builds a high-throughput backscatter network that operates in the mmWave frequency bands. It is able to transmit packets with 1 Gbps and 100 Mbps at 4.6 meters and 8 meters, respectively. Its power consumption is as low as 2.4 nJ/bit. RetroTurbo [36] takes visible light to transmit tag data, which shows an 8 kbps visible light backscatter communication (VLBC) link reliably within 7.5 meters.

Modulation technologies The state-of-art (SoA) backscatter systems adopt various modulation technologies to transmit bit stream ([39], [40], [2], [19], [37], [41], [42], [43], [44]). HitchHike [40] and FreeRider [2] take codeword translation to modulate tag bits. Their key technology is to translate one codeword into another valid codeword from the same codebook. Its effectiveness has been demonstrated in ambient 802.11b, BLE, and ZigBee signals. TScatter [19] implements a novel OFDM backscatter system that uses a high-granularity sample-level modulation to transmit tag bits. Aloba [37] revisits the ON-OFF Keying (OOK) application for LoRa backscatter. Its throughput achieves 39.5-199.4 kbps at various distances. [41] demonstrates how to implement an OFDMA backscatter for Wi-Fi transmissions. It takes the 802.11g framework to validate the system design, which allows 48 tags to transmit bit stream concurrently.



(a) RSSI (b) Throughput

Fig. 17. The distribution of backscatter RSSI and throughput.

VII. Conclusion

BumbleBee is an ambient ZigBee backscatter system that leverages productive BLE signals for RF carriers. It dominates the backscattered signal while the carrier information is negligible. Since BLE signals are widespread, we believe that BumbleBee enables the vision of pervasive ZigBee backscatter communication.

VIII. Acknowledgments

We thank the PerCom reviewers for their helpful comments. This work was supported by NSFC Grant No. 61932017 and 61971390.

References

[1] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith, "Inter-technology backscatter: Towards internet connectivity for implanted devices," in *Proc. of ACM SIGCOMM*, 2016.

[2] P. Zhang, C. Josephson, D. Bharadia, and S. Katti, "Freerider: Backscatter communication using commodity radios," in *Proc. of ACM CONEXT*, 2017.

[3] W. Gong, S. Chen, J. Liu, and Z. Wang, "Mobirate: Mobility-aware rate adaptation using phy information for backscatter networks," in *Proc. IEEE INFOCOM*, 2018.

[4] W. Gong, L. Yuan, Q. Wang, and J. Zhao, "Multiprotocol backscatter for personal iot sensors," in *Proc. ACM CoNEXT*, 2020.

[5] W. Gong, S. Chen, and J. Liu, "Towards higher throughput rate adaptation for backscatter networks," in *Proc. IEEE ICNP*, 2017.

[6] J. Zhao, W. Gong, and J. Liu, "Towards scalable backscatter sensor mesh with decodable relay and distributed excitation," in *Proc. ACM MobiSys*, 2020.

[7] Z. Xu and W. Gong, "Enabling zigbee backscatter communication in a crowded spectrum," in *Proc. IEEE ICNP*, 2022.

[8] W. Gong, I. Stojmenovic, A. Nayak, K. Liu, and H. Liu, "Fast and scalable counterfeits estimation for large-scale rfid systems," *IEEE/ACM Transactions on Networking*, vol. 24, pp. 1052–1064, 2015.

[9] W. Gong, J. Liu, and Z. Yang, "Fast and reliable unknown tag detection in large-scale rfid systems," in *Proc. ACM MobiHoc*, 2016.

[10] J. Yu, J. Liu, R. Zhang, L. Chen, W. Gong, and S. Zhang, "Multi-seed group labeling in rfid systems," *IEEE Transactions on Mobile Computing*, vol. 19, pp. 2850–2862, 2019.

[11] K. Liu, Q. Ma, W. Gong, X. Miao, and Y. Liu, "Self-diagnosis for detecting system failures in large-scale wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 13, pp. 5535–5545, 2014.

[12] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith, "Passive wi-fi: Bringing low power to wi-fi transmissions," in *Proc. of USENIX NSDI*, 2016.

[13] J. F. Ensworth and M. S. Reynolds, "Ble-backscatter: Ultralow-power iot nodes compatible with bluetooth 4.0 low energy (ble) smartphones and tablets," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, pp. 3360–3368, 2017.

[14] Z. Chi, X. Liu, W. Wang, Y. Yao, and T. Zhu, "Leveraging ambient lte traffic for ubiquitous passive communication," in *Proc. ACM SIGCOMM*, 2020.

[15] Z. Xu, Y. Ding, and W. Gong, "A fine-grained modulation technology for zigbee backscatter communication," in *Proc. IEEE MMS*, 2022.

[16] J. Zhao, W. Gong, and J. Liu, "X-tandem: Towards multi-hop backscatter communication with commodity wifi," in *Proc. ACM MobiCom*, 2018.

[17] X. Guo, L. Shangguan, Y. He, N. Jing, J. Zhang, H. Jiang, and Y. Liu, "Saiyan: Design and implementation of a low-power demodulator for {LoRa} backscatter systems," in *Proc. USENIX NSDI*, 2022.

[18] M. Dunna, M. Meng, P.-H. Wang, C. Zhang, P. Mercier, and D. Bharadia, "Syncscatter: Enabling wifi like synchronization and range for wifi backscatter communication," in *Proc. USENIX NSDI*, 2021.

[19] X. Liu, Z. Chi, W. Wang, Y. Yao, P. Hao, and T. Zhu, "Verification and redesign of OFDM backscatter," in *Proc. USENIX NSDI*, 2021.

[20] "Bluetooth Core Specifcation," 2019, https://www.bluetooth.com/specifcations/bluetooth-core-specifcations.

[21] F. Li, "Multi-engine multi-level simulation for system specification validation and power consumption optimization," Ph.D. dissertation, Université Nice Sophia Antipolis, 2016.

[22] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–320, 2006.

[23] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, D. Fang, and K. Jamieson, "PLoRa: A passive long-range data network from ambient LoRa transmissions," in *Proc. of ACM SIGCOMM*, 2018.

[24] "CC2640R2F datasheet," https://www.ti.com/product/CC2640R2F.

[25] "HackRF," https://www.greatscottgadgets.com/hackrf/.

[26] "CC2530 datasheet," https://www.ti.com/product/CC2530.

[27] "CC1352 datasheet," https://www.ti.com/product/CC1352R.

[28] "CC2540 datasheet," https://www.ti.com/product/CC2540.

[29] "CC2650 datasheet," https://www.ti.com/product/CC2650.

[30] "ATMEGA256RF2," https://www.microchip.com/en-us/development-tool/ATMEGA256RFR2-XPRO.

[31] "AD8313 datasheet," https://www.analog.com/cn/products/ad8313.html.

[32] "ADG902 datasheet," https://www.analog.com/en/products/adg902.html.

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

[34] A. Galisteo, A. Varshney, and D. Giustiniano, "Two to tango: Hybrid light and backscatter networks for next billion devices," in *Proc. ACM MobiSys*, 2020.

[35] M. H. Mazaheri, A. Chen, and O. Abari, "Mmtag: A millimeter wave backscatter network," in *Proc. ACM SIGCOMM*, 2021.

[36] Y. Wu, P. Wang, K. Xu, L. Feng, and C. Xu, "Turboboosting visible light backscatter communication," in *Proc. ACM SIGCOMM*, 2020.

[37] X. Guo, L. Shangguan, Y. He, J. Zhang, H. Jiang, A. A. Siddiqi, and Y. Liu, "Aloba: Rethinking on-off keying modulation for ambient lora backscatter," in *Proc. ACM SenSys*, 2020.

[38] J. de Winkel, V. Kortbeek, J. Hester, and P. Pawełczak, "Battery-free game boy," *Proc. ACM IMWUT*, 2020.

[39] A. Wang, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota, "FM backscatter: Enabling connected cities and smart fabrics," in *Proc. of USENIX NSDI*, 2017.

[40] P. Zhang, D. Bharadia, K. Joshi, and S. Katti, "Hitchhike: Practical backscatter using commodity wifi," in *Proc. of ACM SenSys*, 2016.

[41] R. Zhao, F. Zhu, Y. Feng, S. Peng, X. Tian, H. Yu, and X. Wang, "Ofdma-enabled wi-fi backscatter," in *Proc. ACM MobiCom*, 2019.

[42] M. Hessar, A. Najafi, and S. Gollakota, "NetScatter: Enabling Large-Scale backscatter networks," in *Proc. NSENIX NSDI*, 2019.

[43] F. Zhu, Y. Feng, Q. Li, X. Tian, and X. Wang, "Digiscatter: Efficiently prototyping large-scale ofdma backscatter networks," in *Proc. ACM MobiSys*, 2020.

[44] J. Zhao, W. Gong, and J. Liu, "Spatial stream backscatter using commodity wifi," in *Proc. ACM MobiSys*, 2018.