Reliable and Practical Bluetooth Backscatter With Commodity Devices

Si Chen, *Student Member, IEEE*, Maolin Zhang, *Graduate Student Member, IEEE*, Jia Zhao, *Student Member, IEEE*, Wei Gong, *Member, IEEE*, and Jiangchuan Liu, *Fellow, IEEE*

***Abstract*— Recently backscatter communication with commodity radios has received significant attention since specialized hardware is no longer needed. The state-of-the-art BLE backscatter system, FreeRider, realizes ultra-low-power BLE backscatter communication entirely using commodity devices. It, however, suffers from several key reliability issues, including unreliable two-step modulation, productive-data dependency, and lack of interference countermeasures. To address these problems, we propose RBLE, a robust BLE backscatter system that works with an excitation BLE device and a single BLE receiver. First, it uses BLE signals with partial single tones as excitations, making single-bit modulation much more robust. Then it designs dynamic channel configuration that enables channel hopping to avoid interfered channels. Moreover, it presents BLE packet regeneration that uses adaptive encoding to further enhance reliability for various channel conditions. The prototype is implemented using TI BLE radios, iPhones, Android phones, and customized tags with FPGAs. Empirical results demonstrate that RBLE achieves more than 17x uplink goodput gains over FreeRider under indoor LoS, NLoS, and outdoor environments. We also show that RBLE can realize uplink ranges of up to 25 m for indoors and 56 m for outdoors.**

***Index Terms*— Backscatter, battery-free communication, Bluetooth.**

I. Introduction

BACKSCATTER communication has attracted much attention for Internet-of-Things (IoT) applications because it is able to harvest energy from RF sources and provide connectivity to ultra-low-power sensors [1]–[14]. The most popular backscatter communication is the RFID technology that has been widely used in supply chains, asset tracking, and inventory management. Nevertheless, the requirement of specialized and expensive readers has long bedevilled this near zero-power technology's wider adoption. Therefore, a bunch of new backscatter paradigms that work with commodity radios have been proposed recently [15]–[23].

Thanks to widespread Bluetooth radios in our daily life, e.g., smartphones, speakers, and headphones, Bluetooth based backscatter systems have received ever-increasing interest for embedded electronics. FS-backscatter [1] can successfully demodulate backscattered BLE (Bluetooth Low Energy) signals, but requires hacking into a specific chip and thus is not a general solution. Interscatter [24] novelly presents how to backscatter BLE signals into Zigbee and WiFi but fails to interface with Bluetooth receivers. BLE-backscatter [25] introduces how to produce BLE backscatter signals using a dedicated continuous wave (CW) generator. In short, none of the above provides a backscatter solution that is completely built by commercial Bluetooth radios. The most recent work, FreeRider [26], for the first time realizes this goal with only commodity BLE radios. It, however, suffers from several key reliability issues.

1) *Unreliable two-step modulation*. To be compatible with BLE signals, FreeRider employs two-step modulation. It first shifts the exciting signal to the target channel, e.g., 6 MHz frequency shift, and then uses an additional frequency shift, e.g., 500 kHz, to do codeword translation. Detailed later in Section III, doing so would inevitably introduce unreliability due to self-interference or no solid signal in the target frequency. Hence, a new modulation scheme that does not rely on codeword translation is needed to fix this problem.
2) *Productive-data dependency*. Besides the unreliable modulation, FreeRider has another serious issue that it requires the data sequence of exciting signals to decode the tag data, which means if the data sequence of the original channel is corrupted, there is no way to successfully decode the tag data even when the backscatter data sequence is error-free. Though working with productive data is a nice property, such productive-data dependency would significantly impact the BER of the tag data, especially when quality of the original channel becomes unstable due to mobility or occlusion.
3) *Lack of interference countermeasures*. None of the previous Bluetooth based backscatter systems has provided proper countermeasures to interference. In fact, those systems do not perform well in the presence of overlapping channel interference caused by other wireless technologies, including WiFi, ZigBee, cordless phone, microwave oven, which simultaneously work on the crowded 2.4 GHz ISM band.

To address the above issues, we propose RBLE, a robust BLE backscatter system that works with an excitation BLE

Manuscript received February 23, 2020; revised November 27, 2020 and March 22, 2021; accepted March 23, 2021; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor F. Dressler. Date of publication April 1, 2021; date of current version August 18, 2021. This work was supported by NSFC under Grant 61932017 and Grant 61971390. *(Corresponding author: Jiangchuan Liu.)*

Si Chen, Jia Zhao, and Jiangchuan Liu are with the School of Computing Science, Simon Fraser University, Burnaby BC V5A 1S6, Canada (e-mail: sca228@sfu.ca; zhaojiaz@sfu.ca; jcliu@sfu.ca).

Maolin Zhang and Wei Gong are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: zmaolin@mail.ustc.edu.cn; weigong@ustc.edu.cn).

Digital Object Identifier 10.1109/TNET.2021.3068865

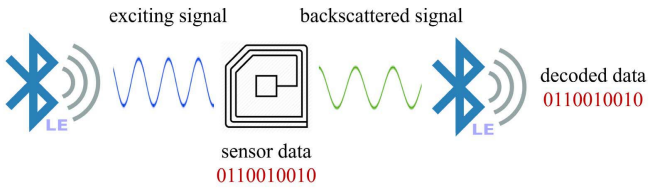


Fig. 1. RBLE conceptual design. The RBLE tag modulates its sensor data on BLE exciting signals and backscatters new BLE packets that any commodity BLE device can decode.

device and a single BLE receiver. As shown in Figure 1, upon receiving signals from excitation BLE devices, the RBLE tag modulates sensor data onto them and backscatters new BLE packets. The whole system involves only commodity BLE devices and RBLE tags. To make BLE backscatter reliable, RBLE makes the following technical contributions:

1) RBLE uses BLE signals with partial single tones as excitations and finds the optimal modulation index, enabling robust BLE packet regeneration bit-by-bit.
2) RBLE designs dynamic channel configuration that allows RBLE tags to perform channel hopping to bypass seriously interfered channels while none of the previous BLE based backscatter systems has provided proper countermeasures to overlapping channel interference.
3) RBLE presents BLE packet regeneration that uses adaptive encoding to further enhance backscatter reliability for challenging situations, e.g., low SNRs. The introduced adaptive encoding for BLE backscatter significantly reduces BERs.

We prototype RBLE using TI CC2540 radios and customized tags implemented by FPGAs. Through extensive empirical evaluation, our main results are summarized as follows.

1) Compared to FreeRider, RBLE achieves more than 17.3x and up to 78.8x goodput gains in LoS cases, and achieves more than 17.1x and up to 66.x goodput gains in NLoS scenarios.
2) The maximum goodput RBLE can achieve is 16.6 kbps, which is 95% of the theoretical capacity for a single excitation commodity BLE radio. Also, the maximum uplink ranges of RBLE are 25 m for indoors and 56 m for outdoors.
3) By the help of dynamic channel configuration, RBLE with channel hopping achieves 1.92x goodput gain over the one without such help in the presence of strong WiFi interference.
4) Our adaptive encoding can significantly reduce BERs. When the uplink distance is 5 m, the BER can be reduced from 0.56% using M1 encoding to 0.1% using M8, and when the uplink distance increases to 20 m, the BER drops from 9.2% using M1 to 0.45% using M8.
5) We also implement RBLE with only off-the-shelf phones, including iPhones and Android phones. The experiments show that RBLE is able to work with both data and advertise packets from smartphones as carriers, and the maximal uplink range is 16 m for an iPhone as the receiver.

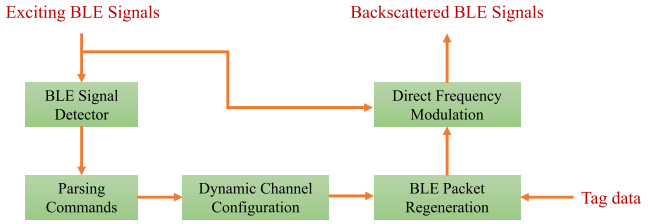


Fig. 2. RBLE framework. After the RBLE tag detects exciting signals, commands are parsed using packet length demodulation. Then it drives dynamic channel configuration module together with tag data to regenerate corresponding BLE signals using direct frequency modulation.

II. BLE Primer

BLE is a massive overhaul of the previous Bluetooth specifications by providing significantly power saving for peripheral devices that do not require high data rates but the lengthened battery life, such as healthcare and fitness applications. It achieves huge market success and thus most operating systems including iOS, Android, as well as macOS, Linux, Windows 8 and 10, natively support BLE [27]–[30]. BLE operates at 2.4 GHz, the unlicensed ISM band and has 1 Mbps raw data rate. It has 40 pre-defined channels ranging from 2400 to 2483.5 MHz, each of which is 2 MHz wide. To build connections and transmit data, the specification defines two kinds of packets: advertising and data packets. BLE devices send advertising packets to broadcast data and to allow other devices to find and connect to them. The BLE device advertises on three channels, channel 37 (2402 MHz), channel 38 (2426 MHz), and channel 39 (2480 MHz).

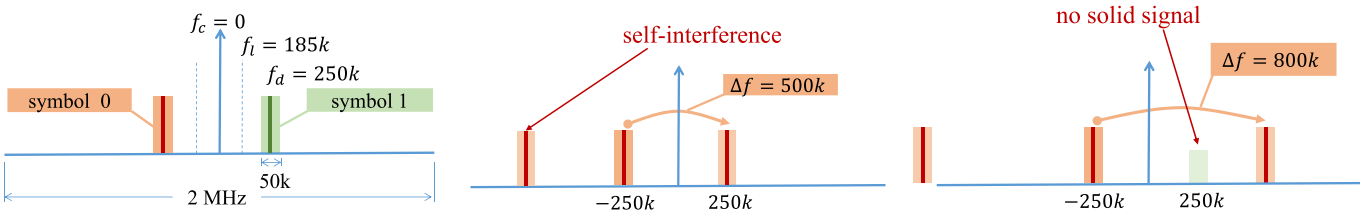
III. RBLE Design

A. Overview

Figure 2 shows the framework of RBLE. A BLE device generates exciting BLE signals to control RBLE tags. Upon BLE signals detected, the RBLE tag parses commands including hopping sequence, channel dwelling time, encoding coefficient, etc. Those core command parameters drive a state machine to dynamically configure channels. Those channel parameters together with tag data are used for BLE packet regeneration to produce raw binary bits, and to decide how to modulate the exciting BLE signals.

B. Modulation Using Direct Frequency Shift

The core of RBLE is how to modulate an exciting BLE signal into another BLE signal. Although we share the same motivation as the seminal work, FreeRider, our solutions are quite different. Before the discussion of how well FreeRider performs, let us briefly review how original BLE signals modulate. As shown in Figure 3(a), in a channel of 2 MHz, the symbol 1 is represented by a position frequency deviation f_d, 250 kHz, and the symbol 0 is represented by a negative frequency deviation of the same amount, following Gaussian frequency shift keying (GFSK). According to BLE specifications, there are two mandatory tests to make sure that first, such deviations should be within 225 kHz and 275 kHz, which



(a) BLE GFSK modulation (b) FreeRider: Case 1 (c) FreeRider: Case 2

Fig. 3. The BLE specifications require a symbol 0 to be encoded as a negative frequency deviation around 250 kHz and a symbol 1 as a positive 250 kHz deviation, shown in (a). When FreeRider does codeword translation, it makes two cases. Case one that choosing frequency shift as 500 kHz to translate 0 to 1 as in (b) brings self-interference caused by the unwanted copy at −750 kHz. Case two is to move this unwanted copy outside the channel by performing frequency shift more than 750 kHz as in (c), leaving no solid signal at 250 kHz. Both cases lead to unreliable modulation for FreeRider.

is 50 kHz wide, and second, at least 99.9% of all deviations must be greater than f_l, 185 kHz.

In order to make backscattered signals still BLE legitimate, FreeRider novelly proposes codeword translation, which translates one codeword to another. Note that this codeword translation happens after a frequency shift that moves the exciting signal from the original channel to the target channel. So we call it two-step modulation. While this proposal works really well with phase modulation for WiFi signals, some unexpected issues arise with GFSK modulation. As in GFSK, there are only two codewords, 0 and 1, which means codeword translation has to encode information by changing 0 to 1 or 1 to 0. In particular, such requirement of GFSK based codeword translation leaves FreeRider only two choices, which is a dilemma. The first choice is shown in Figure 3(b), where the original symbol is 0 and FreeRider wants to make it 1. So it directly performs a frequency shift, $\Delta f = 500$ kHz, then the shifted symbol becomes 1. However, this operation inevitably produces another unwanted copy at $-250-500 = 750$ kHz, which is still within the BLE channel, creating self-interference. As said in FreeRider [26], single side-band cancellation solutions do not apply here because codeword translation is not aware of the original symbol and thus does not have any idea that which side should be cancelled. Hence, the other choice left for FreeRider is to move this unwanted copy out of the channel, then built-in filters on the BLE receiver would ignore it, as depicted in Figure 3(c). So for the same purpose translating 0 to 1, Δf has to be more than 750 kHz. In this case, it indeed moves the unwanted copy signal out of the channel but leaves no solid signal at 250 kHz deviation, where the BLE receiver looks for. In other words, case 2 makes translated codeword susceptible to noise. From the above, we can see that both choices result in unreliable modulation for FreeRider.

We empirically test FreeRider for various Δf and find that for single-bit modulation, BERs are all above 30% when Δf is between 500 kHz and 1 MHz, which presents huge challenges for practical applications. By investigating this problem in depth, our direct frequency-shift works as follows. First, we make the single tone part of a BLE signal as the modulation carrier. Inspired by Interscatter [24], by properly performing reversely-whitening techniques, the payload of the BLE signal can be made all ones or zeros, which are single tones. Next, we directly apply frequency shifts to modulate 0 or 1 in the target channel.

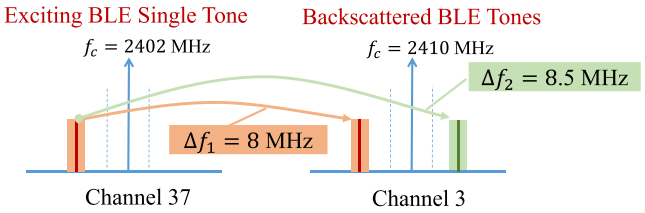


Fig. 4. Direct frequency shift modulation example. To modulate a symbol 0 to target channel 3, we only need to shift a frequency of 8 MHz on the zeros single tone of the exciting signal on channel 37. A frequency shift of 8.5 MHz is for the symbol 1.

As shown in Figure 4, suppose we have all zeros single tone in advertising channel 37, and our target channel is data channel 3. If we need to modulate a symbol 0, a frequency shift of 8 MHz, Δf_1, would be chosen to modulate. Similarly, a frequency shift of 8.5 MHz, Δf_2, would be able to modulate a symbol 1. While such a design is simple, it is efficient and easy to implement. Also, it removes the productive-data dependency brought by FreeRider and requires only a single BLE receiver to decode. Moreover, while the Cyclic Redundancy Check (CRC) result of FreeRider packets is always erroneous, ours can be made right, which is important when the receiver is a smart device supporting BLE because many application software cannot display CRC-error packets [31]–[33]. In addition, for the unwanted copy of our frequency shift, though it would never fall into our target channel, we can still either apply single side-band cancellation techniques [24], [34] or just rely on the filters on the BLE receiver to take care of it, reducing unnecessary interference for other channels.

C. Dynamic Channel Configuration

Most existing backscatter systems [24], [26], [34] are all static, which means targeting at a single channel. Meanwhile, those systems are working on the same 2.4 GHz ISM band, which is quite crowded and thus full of interference. So, we intend to design a dynamic channel configuration scheme to enable channel hopping for RBLE and to reduce interference impact. To do so, first, we need a configurable clock generator that produces two different clocks for our modulation. Hence, we design a Phase Locked Loop (PLL) based clock generator as shown in Figure 5. A phase locked loop is a negative feedback control circuit that utilizes the voltage generated by phase synchronization to tune the voltage

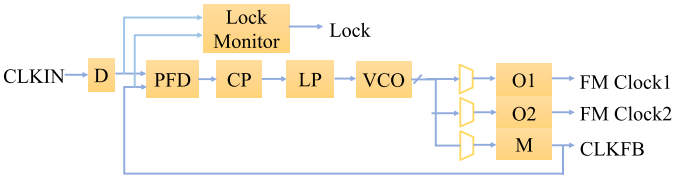


Fig. 5. Configurable clock generator providing two clocks for modulation.

controlled oscillator to produce a target frequency. The clock generator circuit mainly consists of a voltage-controlled oscillator (VCO), a phase frequency detector (PFD), a charge pump (CP), a loop filter (LP) and a lock monitor. The heart of clock generator is the VCO. The VCO outputs a signal, part of which is divided by the input frequency to generate clock1 and clock2. The other part (CLKFB) is phase-compared with the reference signal in PFD to realize feedback; therefore the VCO can output a stable clock signal. O1, O2, M, and D are programmable frequency dividers with configuration registers which adapt the VCO to our clock generator. By configuring these registers, clock generator can generate the required frequencies. The VCO frequency can be determined by:

$$F_{VCO} = F_{CLKIN}(\frac{M}{D}) \tag{1}$$

$$F_{VCOMIN} \leq F_{VCO} \leq F_{VCOMAX} \tag{2}$$

Frequency of two frequency modulation (FM) clocks can be calculated using this formula:

$$F_{FM} = F_{CLKIN}(\frac{M}{D \cdot O}) \tag{3}$$

where F_{VCOMIN} and F_{VCOMAX} represent the controllable frequency range of the VCO, and F_{CLKIN} is the frequency of CLKIN. The M, D, O1, and O2 values come from configuration registers of programmable frequency dividers. The values of M, D must be chosen appropriately to keep the VCO within its frequency range. Based on two generated clocks, we can feed them into the modulation module that uses different clocks to control the RF switch accordingly.

To enable channel hopping, we need multiple sets of different clocks. One way is to generate multiple sets of clocks at the same time, and select one set for output at a time. Nevertheless, it inevitably boosts power consumptions if too many clocks are involved. For example, there are 40 channels for BLE, which requires 80 clocks for hopping across them. To get around of this, we design a dynamic reconfiguration technique to produce required two clocks for each hopping. Thanks to our previous configurable clock generator design, essentially dynamic reconfiguration only needs to vary the voltage level of VCO and reconfigures the registers of programmable frequency dividers. Specifically, dynamic reconfiguration is to dynamically change values of M, D, O1, and O2 in our configurable clock generator.

Our dynamic reconfiguration is performed through a reconfiguration port which provides access to the configuration bits stored in configuration registers. We design a state machine to drive the reconfiguration port. The state machine is used to generate control and data signals for the reconfiguration

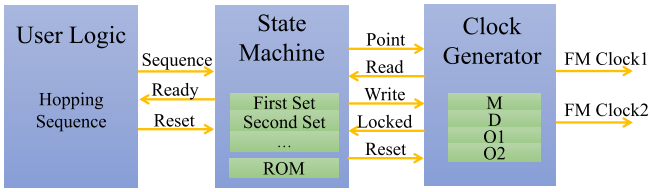


Fig. 6. Channel hopping happens from user logic that parses hopping commands from exciting signals and then tells a state machine where to hop. Upon receiving this command, the state machine controls the reconfigurable clock generator through a set of ports, including Point, Read, Write, Locked, and Reset. After the generator state is locked, it produces two different clocks for the target channel.

port using pre-computed values stored in ROM. This state machine ensures the configuration registers in clock generator are controlled and reconfigured in the correct sequence. Specifically, after the state machine receives the hopping signal from user logic, it first points to the configuration register of the clock generator through the reconfiguration port and then reads the previously configured parameters of the output frequency. Next, it masks the range of bits that need to be changed in configuration registers and chooses the appropriate ROM address according to the reconfiguration state. After that, it writes the updated values of M, D, O1, and O2 to configuration registers. Finally, it waits for the locked signal from the clock generator which indicates the completion of this reconfiguration event. When the frequencies of two clocks are changed, the state machine becomes ready again for the next reconfiguration. The configuration parameters including the addresses, masks, and new configuration values, are stored in a pre-initialized ROM. Figure 6 shows the block diagram of our channel hopping process. The state machine provides multiple reconfiguration states for user logic. The first set is the default state. Other sets correspond to user-configuration loaded into the configuration registers. Each state has a set of pre-computed values of M, D, O1, and O2. The user logic is generated by parsing commands from exciting signals, e.g., hopping sequence.

D. Packet Regeneration

After we have direct frequency shift modulation and channel hopping support, it comes to BLE packet regeneration. Suppose that we already have an exciting advertising packet where the Adv payload is single tone using reversely-whitening techniques [24]. To backscatter a legit packet with tag data, we have two options. First, we backscatter it to another advertising packet in a different advertising channel. In this case, we only need to modulate the tag data onto the Adv Payload field and keep all the fields before it unchanged. The second case is we can backscatter the exciting packet into a data packet. To do so, we need to fabricate the Preamble, Access Address, and Header for the regenerated data packet, and modulate tag data in the Data Payload field. By enabling regenerating both advertising and data packets, RBLE can hop freely and communicate across all 3 advertising channels and 37 data channels. To enable modulating on the single tone part of the signal, we need to find the starting position of the Adv Payload. Particularly, we employ an RF signal power

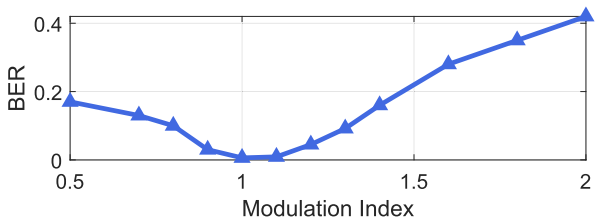


Fig. 7. Adjustment of modulation index. We empirically test RBLE for various modulation indexes and found that when the modulation index is set to 1, the system has the lowest BER.

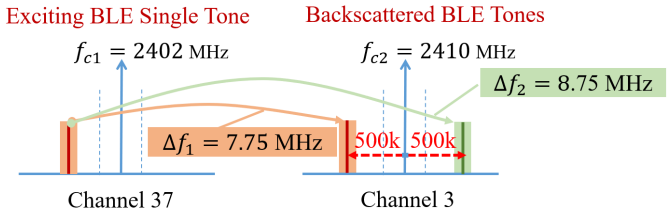


Fig. 8. Direct frequency shift modulation example with a modulation index of 1. In this case, we set the parameter like this: $|\Delta f_2 - \Delta f_1|/2 = 500\ kHz$; $|\Delta f_2 + \Delta f_1|/2 = |f_{c2} - f_{c1}|$.

detector and a voltage comparator. There are lots of off-the-shelf solutions. For example, AD8313 can convert an RF signal to an equivalent DC voltage with high accuracy, and has 40ns signal response time, which fits BLE applications. After the detector discovers a BLE signal, we can have a positive edge generated by the following voltage comparator. Then it skips 104 μs, which is the length of the preamble, access address, header, and adv address. Our modulation starts right after this at the rate of 1 μs per bit. Although our modulation is quite robust as a single-bit solution, it could become relatively unstable when the channel condition changes fast. To tackle this, we borrow the idea in the C1G2 standard [35]–[37] that uses Miller encoding to achieve tradeoff between channel conditions and data rates. Specifically, RBLE enables 3 other different encoding coefficients: 2, 4, 8, which correspond to 2 μs, 4 μs, 8 μs encoding rates. This way, the tag can adapt different encodings based on channel qualities.

E. Downlink Communication

Control communication is required between the tag and the receiver because the tag and receiver need to be synchronized. The BLE receiver performs channel hopping according to the channel quality. Before the receiver hops to a new receiving channel, it needs to send channel parameters to the transmitter. The transmitter forwards these parameters to the tag through downlink communication and then continuously transmits exciting signals for backscatter until it receives the new parameters from the receiver again. Note that most complexities, e.g., when to hop channels, what kind of encoding should be used, lie at the excitation and receiver BLE devices, the RBLE tag just follows orders from standard BLE devices, similar to the RFID tag design. Upon BLE signals detected, the RBLE tag parses commands including hopping sequence, channel dwelling time, encoding coefficient, etc. Those core command parameters drive a state machine to dynamically configure channels and choose the encoding coefficient for BLE packet regeneration. For downlink communication mainly responsible for disseminating parameters, we adopt Packet Length Modulation (PLM) that is widely used by other state-of-the-art systems [12], [26], [38]. For better efficiency and confusion avoidance, the predefined sequence using PLM is used to trigger the command parsing process.

F. Modulation Index of RBLE

RBLE tag regenerates BLE packets using direct frequency shift modulation. Direct frequency shift is essentially a binary frequency shift keying (BFSK) modulation adapted to backscatter. BFSK uses a pair of discrete frequencies to transmit binary (0s and 1s) information [39].

The modulation index of commodity BLE is 0.5. The modulation index indicates by how much the modulated variable varies around its unmodulated level. The modulation index (M) is given by:

$$M = 2 \cdot f_d / R \tag{4}$$

where R is the symbol rate of BLE signal which is 1 Mbps. f_d is frequency deviation. 0.5 is the smallest BFSK modulation index that can be chosen such that the waveforms for symbol 0 and symbol 1 are orthogonal. To regenerate a valid BLE signal, we first set the parameter like this when performing direct frequency shift modulation:

$$f_d = |\Delta f_2 - \Delta f_1|/2 = 250kHz \tag{5}$$

$$|\Delta f_2 + \Delta f_1|/2 = |f_{c2} - f_{c1}| \tag{6}$$

That is, the RBLE modulation index is set to 0.5, which is the same as commodity BLE transmitters. However, through experiments, we find that when the modulation index is 0.5, the modulation performance is poor. This is due to the fact that the GFSK modulation of commodity BLE is more complex than the modulation scheme of RBLE tag. GFSK supports a modulation index of 0.5, which is not the optimal modulation index for RBLE's BFSK modulation. So we need to find the most suitable modulation index for RBLE. We can easily adjust the modulation index of RBLE by changing Δf_1 and Δf_2 used in direct frequency shift. Since frequency deviation must be less than 1 MHz (Channel bandwidth of BLE is 2MHz), the modulation index should be in the range of 0.5 to 2. We empirically test RBLE for various modulation indexes and found that when the modulation index is set to 1, the lowest BER can be obtained at the receiver, as shown in Figure 7. The selected modulation index (1) ensures a sufficiently reliable modulation performance. In summary, we set the parameter like this when performing direct frequency shift modulation:

$$f_d = |\Delta f_2 - \Delta f_1|/2 = 500kHz \tag{7}$$

$$|\Delta f_2 + \Delta f_1|/2 = |f_{c2} - f_{c1}| \tag{8}$$

The bandwidth of the frequency modulated signal is related to the modulation index. The approximate bandwidth of a frequency modulated signal can be estimated by Carson's bandwidth rule:

$$BW = (M + 1) * 1/T_s \tag{9}$$

where T_s is the symbol duration of BLE (1 μs). M is the modulation index. Setting the modulation index to 1 achieves

(a) BLE transceivers (b) RBLE tag prototype

Fig. 9. The photos of TI CC2540 BLE transceivers and RBLE tag prototype.

the lowest BER for BLE backscatter, but the bandwidth of the backscatter signal is increased accordingly. It is worth noting that this does not affect the reception of the backscatter signal by the BLE receiver because the backscatter signal is still within the receiving channel with a bandwidth of 2 MHz.

IV. Implementation

We build a prototype of RBLE using off-the-shelf BLE radios and customized backscatter tags. The implementation is detailed as follows.

BLE Transceiver: We use TI CC2540 radios for both the excitation device and the receiver as shown in Figure 9(a). It transmits at 1 Mbps raw data rate and supports transmission power at 0 dBm and 4 dBm. The frequency deviation is 250 kHz and the modulation index is 0.5. The reversely-whitened sequence for each channel is computed offline.

RBLE Tag: Our RBLE tag prototype shown in Figure 9(b) has two antennas, one for receiving and the other for backscatter. We use the AD8313 envelope detector connected to the receiving antenna. Its detection delay is measured around 0.4 μs, which is negligible for BLE applications. The backscatter antenna is connected to an ADG902 RF switch and baseband processing is implemented using an XILINX Artix-7. The modulation index is set to 1.

Low-Power Tag Design: The biggest challenge for low-power tag design is the oscillator. Our solution is to employ the ring oscillator from [1], [24] that can generate a 35 MHz clock at the power consumption of 28 μW. We simulated the RBLE tag design using TSMC 65 nm technology and the overall power consumption is around 37 μW. 23 μW is consumed by the 30 MHz clock and 11 μW is needed for the RF switch. All the rest goes to running the control logic and modulation.

Experiment Setup: Our experiments are conducted in indoor environments with line-of-sight and non-line-of-sight deployments. The downlink (transmitter-to-tag) distance is 0.3 m. For line-of-sight deployment, we move the receiver along a straight line to increase the uplink (tag-to-receiver) distance. For non-line-of-sight deployment, we place both the transmitter and tag in the room, then move the receiver in the corridor.

Metric: One of the most performance metrics is the uplink goodput. There are two types of goodput. One is total goodput, which counts all the bits the tag modulates on the single-tone. The other is the payload goodput, which only counts tag data bits transmitted. Note that there is no difference between these two goodput types for adv target channels, because the backscattered adv packet is of the same length as the original exciting packet. But for data target channels, the total goodput is always higher than payload goodput. Unless otherwise specified, we refer payload goodput as goodput in the rest of this paper. Although the random delay of the link layer for advertising events is unknown, we empirically confirm that the maximum advertising packet rate is stable around 70 packets/s. Therefore, the goodput capacity using a single excitation device is about 17.4 kbps.

Competition: We mainly compare our system with FreeRider because it is so far the only backscatter system that works entirely with commodity BLE radios. Our FreeRider implementation is based on the Github codes published by the original author [40]. We set the FreeRider encoding length as 16-symbol for two reasons. The published data [40] uses this setting and for shorter encoding lengths, BERs surge significantly according to our experiments.

V. Evaluation

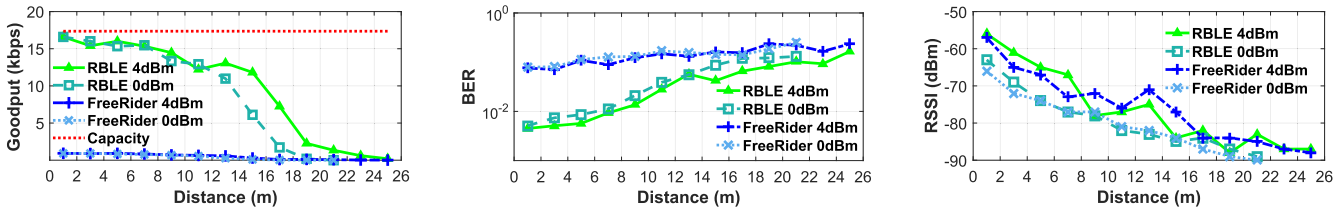
A. Line-of-Sight Deployment

Figure 10 shows the RBLE performance with increasing uplink distances in LoS deployment. The maximum goodput RBLE can achieve is 16.6 kbps. Given that the goodput capacity is 17.4 kbps, our system reaches up to 95% of the theoretical capacity. When the uplink range increases, RBLE is still able to achieve more than 10 kbps rates within 15 m. In addition, backscattered packets of RBLE can be decoded as far as 25 m. Such goodput and coverage are sufficient for many IoT applications.

More importantly, RBLE achieves significant goodput gains over FreeRider for all cases. In particular, RBLE achieves goodput of 15.4 kpbs at the distance of 3m and goodput of 11.8 kbps at 15 m away, which are 17.3x and 40.2x better than counterparts of FreeRider. An interesting observation is that while RSSI gaps are not that obvious for RBLE and FreeRider shown in Figure 10(c), the BER gaps are distinct as in Figure 10(b). This is mainly because our direct frequency shift modulation is much more robust than codeword translation that either causes self-interference or leaves no solid signal at the desired frequency. In other words, our single-bit modulation can achieve much lower BERs than 16-symbol(bit) modulation of FreeRider. We note that the two-step modulation of FreeRider does not cause a significant impact on the power of the backscattered signal. Compared to RBLE, we did not add an analog mixer when replicating FreeRider. We use the FPGA to complete a digital mixing of the clock for codeword translation and the clock for frequency shifting to the target backscatter channel. The intermediate frequency (IF) signal output from the GPIO port of FPGA will be mixed with the exciting signal in the RF switch.

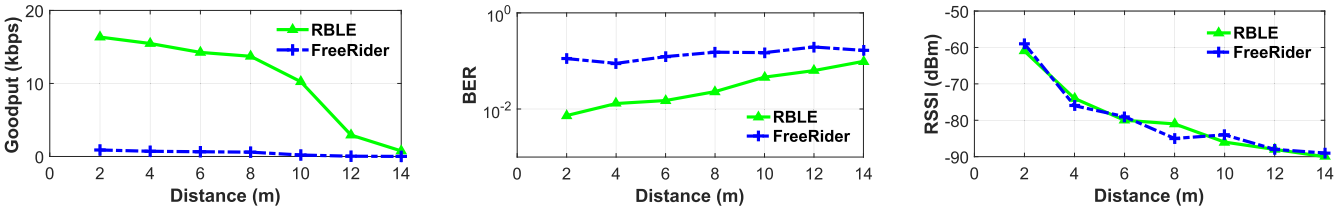
B. Non-Line-of-Sight Deployment

We also investigate the performance of RBLE in the non-line-of-sight scenarios. The exciting signal power is set at 4 dBm and both the transmitter and tag stay in the room while the receiver moves in the corridor. As shown in Figure 11(a), RBLE is still capable of decoding backscattered signals at the



(a) Goodput (b) BER (c) RSSI

Fig. 10. Backscatter goodput, BER, and RSSI across uplink distances in the line-of-sight deployment (downlink distance is 0.3 m).



(a) Goodput (b) BER (c) RSSI

Fig. 11. Backscatter goodput, BER, and RSSI across uplink distances in the non-line-of-sight deployment (downlink distance is 0.3 m).

uplink distance of 14 m. Similar to LoS cases, the maximum achieved goodput is 16.4 kbps at a distance of 2 m. At further distances, RBLE still achieves more than 10 kpbs within 10 m, which confirms that our RBLE is able to deliver decent BLE transmissions for short-range IoT applications, like headphone, smartwatch, and other personal electronics. Figure 11(b) shows that RBLE achieves low BERs even in NLoS cases, and the RSSI degrades to -90 dBm at 14 m so backscatter communication stops as well. This is because when the uplink distance increases more than 14 m, the signal has to penetrate more walls. In consequence, the backscattered signal becomes too weak to decode the packet preamble.

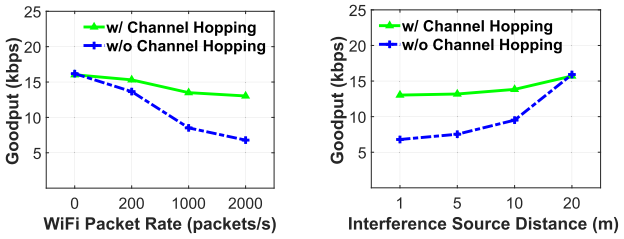
C. Impact of BLE Transmission Power

Next, we evaluate the performance of RBLE when the exciting BLE signals are transmitted at 0dBm and 4dBm. Figure 10(a) shows the goodput results and RBLE again outweighs FreeRider significantly. When the distance is less than 7 m, RBLE's BERs are basically below 1% for both power levels. In contrast, FreeRider's BERs are always above 7% even if it uses 16-symbol encoding, which confirms the superiority of direct frequency shift modulation over two-step modulation of FreeRider. Regarding RSSIs in Figure 10(b), we can see that stronger transmission power brings longer uplink ranges. When the transmission power is 0 dBm, the backscattered packets can be received within 21m, and when the power increases to 4 dBm, the uplink distance reaches 25 m. There is no obvious gap between RBLE and FreeRider.

D. Impact of Channel Hopping

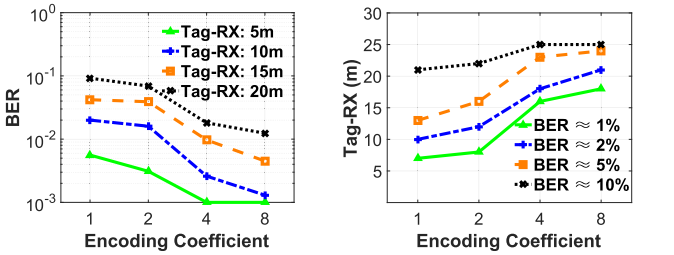
Then, we conduct a set of comparisons to examine the impact of channel hopping in the presence of WiFi interference. We let the interfering WiFi source work at 2412 MHz of 20 MHz wide. The center frequency of the target BLE data channel is 2414 MHz (data channel 5) where this BLE channel is completely within the range of the interfering WiFi channel. Exciting signals are transmitted on adv channel 38. The downlink distance is 0.3 m and the uplink distance is 5 m. We compare performance for two cases, without channel hopping and with channel hopping where the RBLE tag hops across a number of channels set by the exciting BLE signals. We refer to the channel hopping mechanism of active BLE and pre-set a channel hopping sequence on the receiving end. The receiver and tag perform channel hopping according to this sequence. The pre-set channel hopping sequence can not ensure that the tag will jump directly out of the interfered channels, but the tag will perform channel hopping across about 80 MHz. After the channel hopping is activated, the communication time on the interfered channels covered by the interference source will be significantly reduced. In the experiment, we set the receiver to calculate the BER every 10 seconds. If the BER exceeds a pre-set threshold, the receiver will hop to a new receiving channel. Before the hopping, it sends parameters (including the index of the new channel) to the transmitter, which forwards these parameters to the RBLE tag through downlink communication. Since we use one packet to modulate one bit, the downlink data rate is about 50 bps.

Figure 12(a) shows the goodput comparison for those two cases with different WiFi transmission rates. The interfering WiFi source is 1 m away from the BLE receiver. We observe that as the interfering level increases, the goodput of RLE without channel hopping degrades much more than that of RLE with channel hopping. For instance, when the WiFi packet rate increases from 200 to 2000, the goodput without channel hopping drops 6.9 kpbs, from 13.7 kpbs to 6.8 kbps, whereas the goodput with channel hopping only degrades 2.3 kpbs, from 15.3 kbps to 13 kpbs. In other words, RBLE with channel hopping achieves 1.92x goodput gain over the one without channel hopping in the presence of strong WiFi



(a) Different WiFi packet rates. (b) Different WiFi distances.

Fig. 12. Impact of channel hopping on interfering WiFi sources with different WiFi packet rates and different interfering distances.



(a) BER VS Encoding coefficient. (b) Uplink distance VS Encoding coefficient.

Fig. 13. Impact of adaptive encoding on backscatter transmission. Figure 13(a) shows the impact of adaptive encoding on BER for different uplink distances. Figure 13(b) shows to keep the BER at a predefined level, increasing the encoding coefficient can enlarge uplink ranges.

interference. Undoubtedly, such gains are brought by our dynamic channel configuration that enables channel hopping as soon as the BLE receiver discovers there are interfering sources and notifies RBLE tags. Figure 12(b) compares two cases against different interference source distances where the WiFi packet rate is fixed at 2000 packets/s. The results confirm that significant performance gaps exist between RBLE with channel hopping and the one without such. As the interference distance becomes further away, RBLE tends to get closer to the theoretical capacity goodput, 17.4 kpbs. When the WiFi interference source is 20 m away, it basically does not affect our backscatter transmission.

E. Impact of Adaptive Encoding

In order to examine the impact of adaptive encoding on backscatter transmission, we conduct two sets of experiments. Results in Figure 13 demonstrate that with the help of adaptive encoding, BERs can be greatly reduced across a range of different uplink distances. For example, the first group of experiments shown in Figure 13(a) is to examine the effect of adaptive encoding on BER for different uplink ranges. When the uplink distance is 5 m, the BER can be reduced from 0.56% using M1 to 0.1% using M8. Also, when the uplink distance increases to 20 m, the BER drops from 9.2% using M1 to 0.45% using M8. The same observation can be made in the second group of tests shown in Figure 13(b). To keep the BER at a predefined level, increasing the encoding coefficient can enlarge uplink ranges accordingly.

F. Goodput of Adv and Data Channels

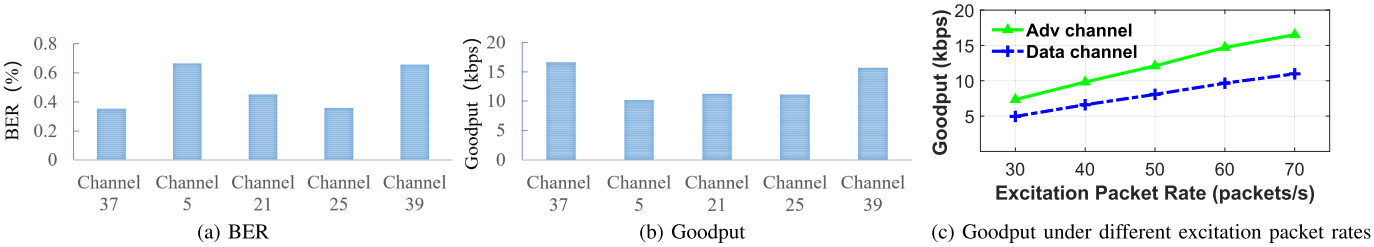
We examine the goodput of our regenerated adv and data packets. We compare the frequency modulation performance (BER) and goodput of five different target channels, which are adv channel 37, data channel 5, data channel 21, data channel 25 and adv channel 39. We set RBLE to hop across these channels. Figure 14(a) shows the performance results of five channels. While the five channels use different frequency shifts, the bit error rates are all less than 1%. Figure 14(b) shows the backscatter goodput of the five channels. We can see that the goodput of data channels is less than that of adv channels. The main reason is that we reuse the preamble, access address, and header of the excitation packet while regenerating advertising packets. In particular, a regenerated adv packet has the payload of 31 bytes while a regenerated data packet has only 21 bytes for the data payload. The same observation can be made from Figure 14(c) as well where we examine goodput differences between Adv and Data channels under different excitation packet rates. We can see that the goodput varies almost linearly with the excitation packet rate.

G. Link Budget

Borrowing the theoretical model for bistatic radar [25], [41], we can estimate RSSIs given link parameters. The received power at the BLE receiver is estimated as $P_R = \frac{P_t G_t \Delta\sigma G_r \lambda^2}{(4\pi)^3 D_1^2 D_2^2}$, where D_1 is the downlink distance. D_2 is the uplink distance. P_t and G_t are the CW source output power and gain of the transmitter's antenna, respectively. G_r is the gain of the receiver's antenna. λ is the carrier-frequency wavelength. $\Delta\sigma$ is the differential radar cross-section (RCS) [42] given by $\Delta\sigma = \frac{\lambda^2}{4\pi} G_N^2 \left|\Gamma_1^* - \Gamma_2^*\right|^2$, where G_N is the antenna gain of the tag and Γ^* is the conjugate match reflection coefficient $\Gamma^* = \frac{Z_a^* - Z_L}{Z_a + Z_L}$ for a resonant antenna impedance Z_a and the complex load impedance Z_L. Z_{L1} and Z_{L2} are measured at the frequency of the exciting signal (2426 MHz in our case), representing impedance of the ADG 902 switch in both on and off states. Values for Z_{L1}, Z_{L2}, and Z_a are measured as $7 - j\ 52\ \Omega$, $12 + j\ 9\ \Omega$, and $50 + j\ 0\ \Omega$ respectively. The estimated RSSI at a variety of uplink distances D_2 is displayed in Figure 15(a). The downward trend of the estimated and measured RSSI is roughly similar. We can also roughly estimate the throughput through simulation, given the backscatter signal strength and the environmental noise floor (-85 dBm). We assume the modulation scheme of the tag is standard BFSK modulation and the demodulation method of the BLE receiver is standard BFSK demodulation. We also assume that the propagation channel is additive white Gaussian noise (AWGN) channel. The estimated throughput at a variety of uplink distances D_2 is displayed in Figure 15(b). The physical layer data rate of BLE is 1 Mbps. SNR decreases as the uplink distance increases. As BER gradually increases, throughput gradually decreases.

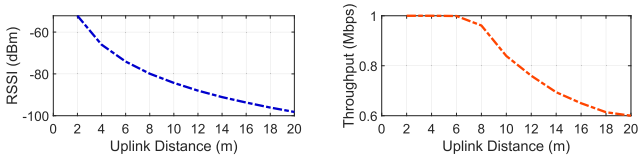
H. Phone-to-Phone Verification

To verify that RBLE is compatible with commodity smart devices with BLE radio, we conduct a phone-to-phone experiment. We use a Huawei P30 and an iPhone 8 Plus as excitation device and receiver respectively. We place the excitation phone at a distance of 0.3 m from the tag. The distance



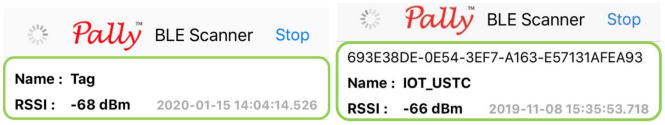
(a) BER (b) Goodput (c) Goodput under different excitation packet rates

Fig. 14. BER and goodput for five different target channels. Figure 14(c) shows goodput differences between Adv and Data channels under different excitation packet rates.



(a) Estimated RSSI (b) Estimated throughput

Fig. 15. Estimated RSSI and throughput at a variety of uplink distances.



(a) BLE advertising packets as excitation. (b) BLE data packets as excitation.

Fig. 16. Generated BLE packets received by an unmodified iPhone. Using BLE advertising packets as excitation, "Tag" can be included in the payload of the regenerated BLE packet. Using longer BLE data packets as excitation, "IOT_USTC" can be included in the payload of the regenerated BLE packet.

between the receiver (iPhone) and the tag is 2 m. We perform reversely-whitening techniques [24] so that the payload of the BLE advertising packet form excitation phone can be made all ones or zeros, which are single tones. The target channel of RBLE tag is advertising channel 38. RBLE tag needs to regenerate a complete BLE packet, including the CRC field. Only when the CRC check result is correct, the smartphone will display the information of the received packet. Figure 16(a) shows the regenerated BLE packets received by an unmodified iPhone with the "BLE Scanner" software. Using BLE advertising packets as excitation, we can only get a single tone no more than 31 bytes. Thus, the length of the whole regenerated packet is no more than 31 bytes, the payload for modulating tag information is less than 21 bytes. RBLE can also be extended to work with BLE data packets. The data packet has a payload of up to 255 bytes. We reversely-whiten BLE data packets using the seed of a specific data channel so that the payload part of the data packet can be used as a longer single-tone to regenerate a BLE packet carrying more information. As shown in Figure 16(b), the name "IOT_USTC" is included in the payload of the regenerated BLE packet while using BLE data packets as excitation.

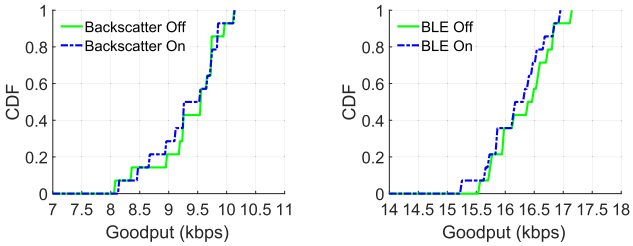
I. Co-Existence With Ambient BLE Transmission

In addition, we investigate the co-existence of our backscatter system with ambient BLE devices. In this experiment, we deploy our backscatter system 1 m away from an ambient BLE transmitter, which transmits advertising packets on adv channel 37 at 40 packets/s. Our backscatter system's excitation sources are advertising packets on adv channel 38. Our RBLE tag backscatters the excitation signal onto adv channel 37. We temporarily stop channel hopping to fully investigate the interaction between the backscatter and ambient BLE signals on the same channel.

Backscatter's Impact on Ambient BLE: Figure 17(a) shows the ambient BLE goodput when our backscatter is present and absent. When we turn on the backscatter transmission, the median BLE goodput is 9.3 kbps. When we turn off the backscatter transmission, the median BLE goodput is 9.5 kbps. Backscatter causes a very slight drop in ambient BLE's goodput, indicating the backscatter communication has no severe interference to the ambient BLE transmission. There are several reasons. First, the backscattered signal strength is usually below -55 dBm, much lower than the signal strength of the ambient BLE transmission. Moreover, our system enables backscatter communication during advertising events, which reduces the time of channel occupancy for backscatter transmissions. In addition, the advertising events have already been perturbed by the random delay, which somehow coordinates the advertising events of different BLE devices. In short, our system does not severely impact ambient BLE devices.

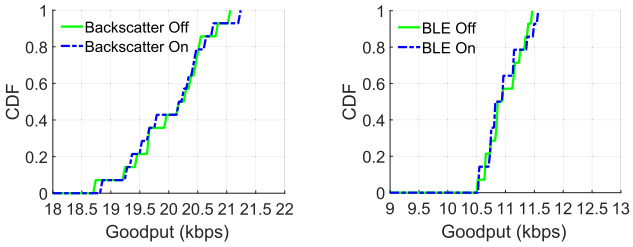
Ambient BLE's Impact on Backscatter: Figure 17(b) shows the backscatter goodput when ambient BLE transmission is present and absent. When we turn off the ambient BLE devices, the median backscatter goodput is 16.4 kbps. When the ambient BLE is turned on, the median backscatter goodput decreases to 16.2 kbps. The ambient BLE's impact on backscatter is slight due to the short channel occupancy time of the backscatter transmission.

We also evaluate the impact of backscatter communication when the backscatter signal is transmitted on a data channel. We deploy our backscatter system 1 m away from two ambient BLE devices which communicate on data channels after establishing a connection. Our RBLE tag backscatters the exciting signal onto data channel 21. As shown in Figure 18 the impact of BLE and backscatter goodput on each other when sharing a data channel is small, even smaller than the impact when sharing an advertising channel. We believe that this is because the time for backscatter and active BLE transmissions to share a data channel is extremely short. After two active BLE



(a) BLE goodput (b) Backscatter goodput

Fig. 17. Impact of BLE and backscatter goodput on each other when sharing an advertising channel.



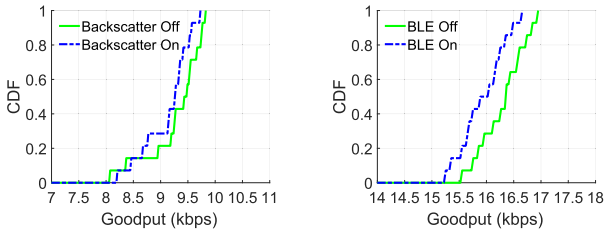
(a) BLE goodput (b) Backscatter goodput

Fig. 18. Impact of BLE and backscatter goodput on each other when sharing a data channel.

devices have established a connection, they will communicate in 37 data channels in a frequency hopping manner, so the time to occupy a specific data channel is short. In addition, the channel occupancy time of the backscatter transmission is also short. As a consequence, the impact of backscatter communication on data-channel BLE transmissions is not obvious. RBLE can coexist well with ambient BLE transmissions on both advertising and data channels.

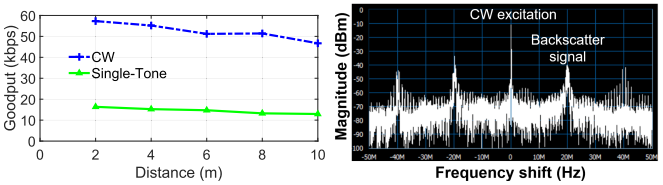
We add two sets of experiments to evaluate the co-existence of backscatter and ambient BLE transmissions on the same advertising channel when increasing the backscatter signal strength and increasing the packet rate of the ambient BLE transmission respectively. For the first set of experiments, we add an amplifier to the excitation device to increase the backscatter signal strength. As shown in Figure 19(a), if we turn on the backscatter transmission, the median BLE goodput drop from 9.5 kbps to 9.25 kbps. The impact is slightly stronger compared to that in Figure 17(a). Relative signal strength impacts the ability of these systems to coexist to a certain extent. For the second set of experiments, we increase the packet rate of the ambient BLE transmission to 70 packets/s, which is the maximum advertising packet rate we can set. As shown in Figure 19(b), if we turn on the ambient BLE transmission, the median backscatter goodput drops from 16.3 kbps to 15.9 kbps. The ambient BLE's impact on backscatter is slightly stronger compared to that in Figure 17(b). The probability of collision impacts the ability of these systems to coexist. However, when ambient BLE is transmitting packets at the maximum rate, the goodput drop caused by ambient BLE is not severely intolerable.

To summarize the above experimental results, when the backscatter signal and the ambient BLE share the same advertising channel, there is no strong mutual interference.



(a) BLE goodput when increasing the backscatter signal strength. (b) Backscatter goodput when increasing the packet rate of BLE.

Fig. 19. Impact of BLE and backscatter goodput on each other when increasing the backscatter signal strength and increasing the packet rate of the ambient BLE transmission respectively (on the same advertising channel).



(a) Backscatter goodput across uplink distances in LoS scenario. (b) Spectrum of the CW excitation and the backscattered signal.

Fig. 20. Backscatter goodput and spectrum when using CW excitation.

In addition, both the ambient BLE devices and our backscatter tags have a channel hopping mechanism, which would further reduce collision chances. Therefore, our backscatter system can coexist well with ambient BLE transmissions.

J. Continuous CW Excitation

In addition to the single tone obtained by setting the application layer data of commodity BLE, we can also use the test mode of commodity BLE chips to generate CW signal. Using CW as excitation, we can freely control the packet generation rate of the RBLE tag. We experiment to compare these two types of excitations. For CW excitation, we set the packet generation rate of the tag to 250 packets/s. For single-tone excitation, the BLE transmitter transmits BLE advertising packets at a rate of 70 packets/s which is consistent with previous experimental settings. The tag's packet regeneration rate is limited by the excitation packet rate. We measure the backscatter goodput of two different types of excitations at the BLE receiver respectively. Backscatter goodput across distances in indoor LoS scenario is shown in Figure 20(a). When the uplink distance is within 10m, the goodput of CW excitation is always above 46 kbps and the goodput of single-tone excitation is always below 17 kbps. Compared to using single-tone excitation, using CW excitation can greatly increase the packet generation rate and thus improve the final backscatter goodput. We have already seen that the goodput varies almost linearly with the excitation packet rate in Figure 14(c). It can be further inferred that the goodput of RBLE is limited by the excitation packet rate. As long as the excitation packet rate is increased, the goodput of RBLE will also increase.

Figure 20(b) shows the spectrum of the CW excitation and the backscattered signal. The backscattered signal (generated BLE signal) is 20 MHz away from the CW excitation.

The RBLE tag was configured to generate BLE advertising packets with a frequency shift of 20 MHz away from the CW excitation. We use a NI PXIe-5663 RF Vector Signal Analyzer as our spectrum analyzer. Note that the backscatter signal has harmonic components. These harmonics have a negative impact on the legacy transmissions on the other channels and we cannot guarantee that harmonics are not generated outside of the 2.4 GHz band. However, the negative impact of harmonics is very small. The energy of the first harmonic is about 10 dB lower than the energy of the main signal, and the energy of the higher harmonics will be lower. We have evaluated the coexistence of the main backscatter signal and ambient BLE transmission in subsection V.I and found that RBLE's main backscatter signal can coexist well with ambient BLE transmissions. The energy of the harmonics is much lower than the energy of the main backscatter signal, so the impact of the harmonics is also limited. Similar to prior systems, RBLE cannot fully eliminate backscatter harmonics, which we intend to investigate in the future.

VI. Related Work

Our work is inspired by recent progress on backscatter systems. The closely related work and corresponding differences are discussed as follows.

WiFi Backscatter: Due to ubiquitous WiFi deployment, researchers have made considerable effort to improve throughput and ranges of WiFi backscatter systems [1], [26], [34], [38], [43]–[45]. Reference [43] attempts to piggyback information on top of WiFi signals at a packet level and achieves 1 kbps and 2.1 m uplink range. Reference [45] makes use of full-duplex radios to largely improve the throughput to 1 Mbps at a range of 5 meters. Reference [44] employs a dedicated CW generator to backscatter 802.11b packets using 4-5 orders of magnitude lower power than normal WiFi. Nevertheless, those systems cannot be built fully with commodity radios. To tackle this challenge, Hitchhike [34] and FreeRider [26] leverage novel codeword translation to build commodity WiFi backscatter systems for 802.11b and 802.11n. MOXcatter further provides the design of how to backscatter spatial stream WiFi [38], and X-Tandem introduces the first multi-hop backscatter paradigm that enables multiple tags to transmit sensor data on the same carrier packet [12], attempting to combine PHY and routing design together; therefore large-scale and high-throughput backscatter networks can be made possible. The modulation schemes of the WiFi backscatter tags [26], [34], [38], [44] are mainly based on Phase Shift Keying (PSK) modulation because 802.11b WiFi uses DBPSK modulation, while 802.11n WiFi uses QAM modulation that combines phase and amplitude modulation.

BLE Backscatter: FreeRider also extends codeword translation to Bluetooth technologies using two-step modulation. As shown in Section III, it suffers from serious problems and cannot make robust modulation. Although FreeRider [26] is the first full BLE backscatter system, it requires two BLE receivers to decode while we only need one. FS-backscatter [1] modulates tag data on Bluetooth exciting signal through amplitude-shift keying (ASK) modulation so it requires hacking into a specific Bluetooth chip to demodulate the amplitude information. Interscatter [24] enables WiFi backscatter through reversely-whitened BLE signals. It proposes reversely-whitening techniques to generate the BLE single-tone. This inspires us to generate a single-tone as the carrier for BLE backscatter. Although our single-tone generation uses reversely-whitening techniques proposed by Interscatter, Interscatter's goal is totally different from ours. Specifically, Interscatter intends to generate WiFi 802.11b signals using DBPSK modulation while we need to regenerate BLE signals using BFSK modulation. Interscatter is a cross protocol backscatter communication system, while RBLE is a BLE backscatter system that works with an excitation BLE device and a single BLE receiver. We are also inspired by [46] that first fine-tunes and applies BFSK modulation to BLE backscatter. BLE-backscatter [25], an expanded version of [46], improves the tag implementation by changing the laboratory instruments including arbitrary waveform generator to MSP430. BLE-Backscatter [25] requires a specialized CW generator while we use commodity BLE devices, reducing the deployment cost of the system. Moreover, the specialized CW generator solution can completely jam a BLE channel and has no downlink capability whereas we employ an envelope detector to identify exciting BLE signals and to receive commands from excitation BLE devices. As far as the modulation part is concerned, the changes we made compared to BLE-backscatter [25] are: changing the carrier generator from specialized device to commodity BLE device and conducting BER performance evaluations to find the optimal modulation index for BLE backscatter. Reference [47] applies BLE-Backscatter [25] to uplink for wireless neural recording. NeuroDisc BLE-compatible backscatter [48] improves BLE Backscatter [25] by introducing single sideband (SSB) backscatter and CPFSK to BLE backscatter. It improves the spectral efficiency of BLE backscatter, but still shares the same limitations as BLE backscatter. Neverthess, none of the above systems had in-depth examined backscatter reliability issues. Relacks [49] proposes a closed-loop configuration selection algorithm that uses frequency, antenna, and wake-up source diversity to deliver reliable transmissions in indoor environments. Relacks implements frequency reconfiguration through channel reconfiguration of BLE transceiver and the tag simply frequency shift by a fixed amount. In this way, the tag does not need to have the capability of dynamic channel configuration. System complexity migrates to the BLE transceiver. Our approach is different. We make full use of the reconfiguration capability of our configurable clock generator to allow the tag to perform dynamic channel reconfiguration. RBLE tag can perform channel hopping following orders, bearing certain system complexity within its capabilities.

In summary, inspired by prior work, we build RBLE using only commodity BLE radios and provide direct frequency shift modulation, dynamic channel reconfiguration, and adaptive encoding techniques to improve backscatter reliability. Extensive empirically evaluations confirm the effectiveness of our design and performance gains over state-of-the-art systems.

VII. Discussion

Productive and Unproductive Exciting Signals: There are trade-offs in choosing an unproductive single tone or a productive data-carrying signal as the carrier. FreeRider requires the data sequence of exciting signals to decode the tag data. If the data sequence of the original channel is corrupted, it is difficult to decode tag data even when the data from the backscattered channel is error-free. Such productive-data dependency would significantly impact the BER of the tag data when the quality of the original channel becomes unstable. In addition, using a productive signal with bandwidth as a carrier will also degrade the BER performance of the modulation.

On the other hand, backscattering with productive exciting signals is a significant step towards exploiting rich ambient signals because it does not need to control the payload content of the exciting signal. We are actively looking for a compromise solution, which can not only use productive exciting signals but also guarantee the reliability of backscatter transmission. One way is to properly control the data content of exciting signals, maintaining a certain amount of productive data goodput.

Choice of Codeword Scheme: The codeword scheme used by RBLE is essentially a repetition code with several different code rates. The error correction capability of the repetition code is limited. We can adopt more complex codeword schemes with stronger error correction capability, such as BCH codes. If we choose BCH codes as the codeword scheme, we could use different code rates of BCH codes to complete the adaptive encoding.

VIII. Conclusion

In this paper, we have proposed RBLE, a reliable BLE backscatter system that works with a single commodity receiver. The main contributions lie in using BLE signals with partial single tones as excitations for BLE backscatter, leveraging dynamic channel reconfiguration to bypass interfered channels, and using adaptive encoding to further improve reliability for challenging low SNR scenarios. Comprehensive field studies demonstrate significant performance gains over state-of-the-art systems in terms of BER, goodput and uplink range. Our future work includes investigation of how to extend RBLE to work with Bluetooth 5.x environments, parallel transmission for multiple BLE tags, and interactions with other protocols, e.g., WiFi and Zigbee.

References

[1] P. Zhang, M. Rostami, P. Hu, and D. Ganesan, "Enabling practical backscatter communication for on-body sensors," in *Proc. ACM SIGCOMM*, 2016, pp. 370–383.

[2] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith, "Ambient backscatter: Wireless communication out of thin air," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 39–50, 2013.

[3] P. Hu, P. Zhang, and D. Ganesan, "Laissez-faire: Fully asymmetric backscatter communication," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 255–267, 2015.

[4] W. Gong, K. Liu, and Y. Liu, "Directional diagnosis for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1290–1300, May 2015.

[5] S. Naderiparizi, M. Hessar, V. Talla, S. Gollakota, and J. R. Smith, "Towards battery-free HD video streaming," in *Proc. USENIX NSDI*, 2018, pp. 233–247.

[6] X. Xu *et al.*, "Passivevlc: Enabling practical visible light backscatter communication for battery-free IoT applications," in *Proc. ACM MobiCom*, 2017, pp. 180–192.

[7] W. Gong, H. Liu, L. Chen, K. Liu, and Y. Liu, "Fast composite counting in RFID systems," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2756–2767, Oct. 2016.

[8] A. Wang, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota, "FM backscatter: Enabling connected cities and smart fabrics," in *Proc. USENIX NSDI*, 2017, pp. 243–258.

[9] P. Hu, P. Zhang, M. Rostami, and D. Ganesan, "Braidio: An integrated active-passive radio for mobile devices with asymmetric energy budgets," in *Proc. ACM SIGCOMM*, 2016, pp. 384–397.

[10] W. Gong, J. Liu, and Z. Yang, "Fast and reliable unknown tag detection in large-scale RFID systems," in *Proc. ACM MOBIHOC*, 2016, pp. 141–150.

[11] O. Abari, D. Vasisht, D. Katabi, and A. Chandrakasan, "Caraoke: An E-toll transponder network for smart cities," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 297–310, Sep. 2015.

[12] J. Zhao, W. Gong, and J. Liu, "X-tandem: Towards multi-hop backscatter communication with commodity WiFi," in *Proc. ACM MobiCom*, 2018, pp. 497–511.

[13] W. Gong, J. Liu, and Z. Yang, "Efficient unknown tag detection in large-scale RFID systems with unreliable channels," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2528–2539, Aug. 2017.

[14] W. Gong, I. Stojmenovic, A. Nayak, K. Liu, and H. Liu, "Fast and scalable counterfeits estimation for large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1052–1064, Apr. 2016.

[15] M. Hessar, A. Najafi, and S. Gollakota, "Netscatter: Enabling large-scale backscatter networks," in *Proc. USENIX NSDI*, 2019, pp. 271–284.

[16] W. Gong, S. Chen, and J. Liu, "Towards higher throughput rate adaptation for backscatter networks," in *Proc. IEEE ICNP*, Oct. 2017, pp. 1–10.

[17] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "LoRa backscatter: Enabling the vision of ubiquitous connectivity," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 1–24, Sep. 2017.

[18] H. Liu, W. Gong, X. Miao, K. Liu, and W. He, "Towards adaptive continuous scanning in large-scale RFID systems," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 486–494.

[19] K. Liu, Q. Ma, W. Gong, X. Miao, and Y. Liu, "Self-diagnosis for detecting system failures in large-scale wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 10, pp. 5535–5545, Oct. 2014.

[20] J. Zhao, W. Gong, and J. Liu, "Towards scalable backscatter sensor mesh with decodable relay and distributed excitation," in *Proc. ACM MobiSys*, 2020, pp. 67–79.

[21] W. Gong, S. Chen, J. Liu, and Z. Wang, "MobiRate: Mobility-aware rate adaptation using PHY information for backscatter networks," in *Proc. INFOCOM*, Apr. 2018, pp. 1259–1267.

[22] V. Talla, B. Kellogg, S. Gollakota, and J. R. Smith, "Battery-free cellphone," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–20, Jun. 2017.

[23] W. Gong, L. Yuan, Q. Wang, and J. Zhao, "Multiprotocol backscatter for personal IoT sensors," in *Proc. ACM CoNEXT*, 2020, pp. 261–273.

[24] V. Iyer, V. Talla, B. Kellogg, J. R. Smith, and S. Gollakota, "Inter-technology backscatter: Towards Internet connectivity for implanted devices," *GetMobile, Mobile Comput. Commun.*, vol. 21, no. 3, pp. 35–38, Nov. 2017.

[25] 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 Trans. Microw. Theory Techn.*, vol. 65, no. 9, pp. 3360–3368, Sep. 2017.

[26] P. Zhang, C. Josephson, D. Bharadia, and S. Katti, "Freerider: Backscatter communication using commodity radios," in *Proc. ACM CONEXT*, 2017, pp. 389–401.

[27] (2020). *BLE API on macOS*. [Online]. Available: https://developer.apple.com/bluetooth

[28] (2020). *BLE API on Ubuntu*. [Online]. Available: https://core.docs.ubuntu.com/en/stacks/bluetooth/bluez/docs/reference/dbus-api

[29] (2017). *BLE API on Windows*. [Online]. Available: https://docs.microsoft.com/en-us/windows/uwp/devices-sensors/bluetooth-low-energy-overview

[30] (2020). *BLE API on Android*. [Online]. Available: https://developer.android.com/guide/topics/connectivity/bluetooth-le

[31] (2020). *nRF Connect*. [Online]. Available: https://apps.apple.com/us/app/nrf-connect/id1054362403

[32] (2020). *LightBlue*. [Online]. Available: https://apps.apple.com/us/app/lightblue/id557428110
[33] (2020). *BLE Scanner*. [Online]. Available: https://apps.apple.com/us/app/ble-scanner-4-0/id1221763603
[34] P. Zhang, D. Bharadia, K. Joshi, and S. Katti, "Hitchhike: Practical backscatter using commodity WiFi," in *Proc. ACM SenSys*, 2016, pp. 259–271.
[35] (2017). *EPC C1G2 Standard*. [Online]. Available: http://www.gs1.org/epcrfid/epc-rfid-uhf-air-interface-protocol/2-0-1
[36] W. Gong *et al.*, "Channel-aware rate adaptation for backscatter networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 751–764, Apr. 2018.
[37] W. Gong, H. Liu, K. Liu, Q. Ma, and Y. Liu, "Exploiting channel diversity for rate adaptation in backscatter communication networks," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.
[38] J. Zhao, W. Gong, and J. Liu, "Spatial stream backscatter using commodity WiFi," in *Proc. ACM MobiSys*, 2018, pp. 191–203.
[39] J. G. Proakis and M. Salehi, *Digital Communications*, vol. 4. New York, NY, USA: McGraw-Hill, 2001.
[40] (2017). *FreeRider Implementation*. [Online]. Available: https://github.com/pengyuzhang/FreeRider
[41] J. D. Griffin and G. D. Durgin, "Complete link budgets for backscatter-radio and RFID systems," *IEEE Antennas Propag. Mag.*, vol. 51, no. 2, pp. 11–25, Apr. 2009.
[42] P. V. Nikitin, K. V. S. Rao, and R. D. Martinez, "Differential RCS of RFID tag," *Electron. Lett.*, vol. 43, no. 8, pp. 431–432, Apr. 2007.
[43] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-Fi backscatter: Internet connectivity for RF-powered devices," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 607–618, Feb. 2015.
[44] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith, "Passive Wi-Fi: Bringing low power to Wi-Fi transmissions," in *Proc. USENIX NSDI*, 2016, pp. 151–164.
[45] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti, "BackFi: High throughput WiFi backscatter," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 283–296, 2015.
[46] J. F. Ensworth and M. S. Reynolds, "Every smart phone is a backscatter reader: Modulated backscatter compatibility with Bluetooth 4.0 low energy (BLE) devices," in *Proc. IEEE RFID*, Apr. 2015, pp. 78–85.
[47] J. Rosenthal, A. Pike, and M. S. Reynolds, "A 1 Mbps 158 pj/bit Bluetooth low energy (BLE) compatible backscatter communication uplink for wireless neural recording in an animal cage environment," in *Proc. IEEE RFID*, Apr. 2019, pp. 1–6.
[48] J. Rosenthal and M. S. Reynolds, "A 1.0-Mb/s 198-pJ/bit Bluetooth low-energy compatible single sideband backscatter uplink for the NeuroDisc brain–computer interface," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 4015–4022, Oct. 2019.
[49] M. Katanbaf, V. Jain, and J. R. Smith, "Relacks: Reliable backscatter communication in indoor environments," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 4, no. 2, pp. 1–24, 2020.

Si Chen (Student Member, IEEE) received the bachelor's degree from the China University of Geosciences and the master's degree from Simon Fraser University, where she is currently pursuing the Ph.D. degree with the School of Computing Science. Her research interests include wireless networks and big data.

Maolin Zhang (Graduate Student Member, IEEE) received the B.S. degree from the Department of Electronic and Information Engineering, Northeastern University, Qinhuangdao, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, University of Science and Technology of China. His advisor is Prof. Wei Gong. His research interests include backscatter communication and wireless networks.

Jia Zhao (Student Member, IEEE) received the M.S. degree in electronic and information engineering from Beijing Jiaotong University, Beijing, China. He is currently pursuing the Ph.D. degree with the School of Computing Science, Simon Fraser University, BC, Canada. His research interests include networking, multimedia communications, cloud computing, and transport protocols.

Wei Gong (Member, IEEE) received the B.S. degree from the Department of Computer Science and Technology, Huazhong University of Science and Technology, and the M.S. and Ph.D. degrees from the School of Software and Department of Computer Science and Technology, Tsinghua University. He is currently a Professor with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include backscatter networks, edge systems, and the IoT applications.

Jiangchuan Liu (Fellow, IEEE) received the B.Eng. degree *(cum laude)* in computer science from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree in computer science from The Hong Kong University of Science and Technology in 2003. He is currently a University Professor with the School of Computing Science, Simon Fraser University, BC, Canada. He is also a NSERC E.W.R. Steacie Memorial Fellow. He was a co-recipient of the Inaugural Test of Time Paper Award of IEEE INFOCOM in 2015, the ACM SIGMM TOMCCAP Nicolas D. Georganas Best Paper Award in 2013, and the ACM Multimedia Best Paper Award in 2012. He has served on the Editorial Boards of IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and IEEE INTERNET OF THINGS JOURNAL. He is also a Steering Committee Member of IEEE TRANSACTIONS ON MOBILE COMPUTING.