Multiprotocol Backscatter With Commodity Radios for Personal IoT Sensors

Longzhi Yuan, *Student Member, IEEE*, Qiwei Wang, *Student Member, IEEE*, Jia Zhao, *Student Member, IEEE*, and Wei Gong, *Member, IEEE*

***Abstract*— We present multiscatter, a novel battery-free backscatter design that can simultaneously work with multiple excitation signals for personal IoT sensors. Specifically, we show for the first time that the backscatter tag can identify various excitation signals in an ultra-low-power way, including WiFi, Bluetooth, and ZigBee. Further, we employ a new modulation approach, overlay modulation, that can leverage those excitation signals to convey tag data on top of productive data, which makes decoding both data possible with only a single personal radio. Moreover, we introduce a low-power listening scheme to improve energy efficiency. Since 2.4 GHz signals and personal radios are everywhere, multiscatter is readily deployable in our everyday IoT applications. We prototype multiscatter using an FPGA and various commodity radios. Extensive experiments show that for mixed 802.11b&n, Bluetooth and ZigBee signals, the average identification accuracy of four protocols is more than 93%. The maximal aggregate throughput of both productive and tag data is 278.4 kbps with a single Bluetooth radio. When the transmitter-to-tag distance is increased from 0.2 to 1.8 m, the maximal communication for BLE drops from 71 m to 29 m. And it can leverage excitation diversity to provide uninterrupted communication and greater throughput gains, whereas the single-protocol tag being idle when carrier signals are unavailable. With indoor office light as harvesting sources, the low-power listening scheme can support backscatter rate at 12 pkts/s.**

***Index Terms*— Backscatter, system, WiFi, BLE, ZigBee.**

I. Introduction

BACKSCATTER communication is one of the most essential technologies for Internet-of-Things (IoT) applications since it can provide ubiquitous connectivity to ultra-low-power sensors [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. A typical backscatter design consists of three key parts: a carrier provider, a backscatter tag, and a backscatter receiver. Taking RFID as an example, the RFID reader plays two roles as the carrier provider and receiver. Despite reduced system complexity by such a dual-role design, the high building cost prevents its mass adoption in personal IoT applications. Because it requires dedicated RFID readers and cannot reuse widely-deployed commodity radios that do not originally support receiving backscatter signals. In consequence, backscatter researchers have made tremendous effort to explore existing signals and commodity radios for backscatter communication in the past decade [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31].

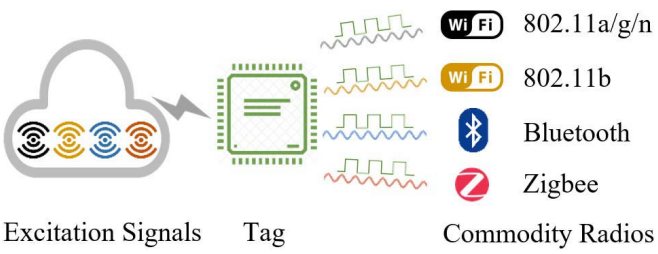


Fig. 1. Multiscatter conceptual design. The multiscatter tag is able to identify different excitation signals and uses overlay modulation to convey tag data with productive data. Only a single commodity personal radio is adequate to decode both data.

So we envision ready-to-use personal backscatter sensors should meet the following requirements:

(a) Universal: It should be able to support excitation diversity, i.e., working with intermittent multiple carriers, such as WiFi, Bluetooth, and ZigBee, so that backscatter connectivity can be significantly improved.

(b) Compatible: It should allow excitation signals to carry productive data and serve as carriers at the same time because non-productive signals are not common and greatly reduce spectral efficiency.

(c) Deployable: It should support single personal-radio decoding for easy and wide adoption since the requirement of more hardware or firmware modification would cause more cost for personal IoT.

The above requirements seem simple, but in fact no backscatter design satisfies all of them for at least two reasons. First, no prior backscatter design investigates how to identify different excitation signals and exploit such diversity. The closest work, FreeRider [30], provides a holistic way to modulate multiple excitation signals, but does not support excitation

Manuscript received 11 January 2021; revised 28 August 2021 and 26 February 2022; accepted 17 September 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. Khreishah. This work was supported by NSFC under Grant 61932017 and Grant 61971390. Part of the results was presented in ACM CoNEXT 2020 [DOI: 10.1145/3386367.3432883]. *(Corresponding author: Wei Gong.)*

Longzhi Yuan is with the School of Data Science, University of Science and Technology of China, Hefei 230027, China (e-mail: longzhi@mail.ustc.edu.cn).

Qiwei Wang is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: qiweiw@mail.ustc.edu.cn).

Jia Zhao is with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: zhaojiaz@sfu.ca).

Wei Gong is with the School of Computer Science and Technology and the School of Data Science, University of Science and Technology of China, Hefei 230027, China (e-mail: weigong@ustc.edu.cn).

Digital Object Identifier 10.1109/TNET.2022.3213913

TABLE I
COMPARISON OF BACKSCATTER SYSTEMS

	Excitation diversity	Productive carrier	Single commodity receiver
WiFi backscatter [23]		●	●
FS backscatter [25]		●	●
Interscatter [22]			●
Passive WiFi [26]			●
LoRa backscatter [28]			●
Hitchhike [29]		●	
FreeRider [30]		●	
X-Tandem [31]		●	
PLoRa [27]		●	
Multiscatter	●	●	●

diversity to distinguish different protocols simultaneously. [32] introduces a backscatter modulation that can work with multiple protocols, but the tag is not able to identify the excitation signals, making a software-defined radio (SDR) necessary to decode tag data. Second, while packet-level backscatter systems, e.g., WiFi backscatter [23] and FS backscatter [25], are compatible and deployable at extremely low data rates, the community shifts focus to symbol-level solutions for higher throughput and better ranges. Unfortunately, a dilemma arises: they have to take sides: either working with non-productive carriers or requiring more (specialized) hardware to do decoding. For example, in interscatter [22] and LoRa backscatter [28], only a single commodity receiver is needed, but the carrier has to be single tones generated by a Bluetooth device. In contrast, Hitchhike [29], FreeRider [30], and X-Tandem [31] can take any productive signals as carriers yet requires two synchronized receivers to decode the tag data. PLoRa [27], though supports productive carriers, cannot work with commodity receivers. A detailed comparison of state-of-the-art backscatter systems is shown in Table I. In short, designing a backscatter system that is universal, compatible, and deployable for personal IoT remains a big challenge.

In this paper, we present multiscatter, a novel backscatter design that satisfies all the above requirements. It works with multiple excitation signals by identifying different protocols first and then modulates data onto the productive carriers accordingly. We observe two unique opportunities for multiscatter: *abundant 2.4 GHz signals* everywhere, home, office, malls, cafes, etc, and *ubiquitous personal radios*, e.g., smartphones, that support WiFi/Bluetooth communications. By reusing existing 2.4GHz excitation signals and pervasive commodity radios as backscatter infrastructure, multiscatter significantly lowers the barrier to wide adoption and realizes readily deployable backscatter communication for our everyday applications. To make this possible, however, we need to address two main challenges.

(a) How to distinguish different excitation packets?

In wireless communication, every packet has a preamble part that defines a specific series of chips to identify itself. Identifying such preambles from high-bandwidth signals on tags, however, is extremely difficult, because unlike active radios, backscatter tags do not have power-hungry components, e.g., amplifier, high-frequency oscillator, to acquire high-bandwidth baseband signals, like WiFi. Further, enabling tags to support multiprotocol identification exacerbates the problem as resources are severely limited for an ultra-low-power design. Our solution is to design a high-bandwidth rectifier that is able to produce high-quality amplitude signals for 802.11b/n, Bluetooth, and ZigBee identification. Such a design is realized by using simple hardware elements, like diodes, capacitors, and resistors. In contrast, prior RFID solutions only support bandwidths of less than 160 kbps. Besides, we employ various techniques together, including quantization, downsampling, and ordered matching, to significantly reduce computation and storage overhead while keeping identification results accurate. The detailed design is described in §II-B, §II-C.

(b) How to modulate productive packets and make them decodable on a single commodity radio?

Backscattering with productive carriers is a significant step towards exploiting excitation signals. The state-of-the-art systems, e.g, Hitchhike, X-Tandem, PLoRa, however, all have to rely on productive data in the original channel to decode tag data, which means if the original productive data is corrupted somehow, there is no way to successfully recover tag data even with error-free backscattered packets. To address this, we propose overlay modulation, a novel modulation approach that modulates tag data on top of ambient signals like "single tone". Specifically, tag modulation is done by creating phase/frequency differences between the reference and modulatable symbols. To decode both productive and tag data, a single commodity radio is enough because reference symbols contain productive data, and comparing them against modulated symbols would recover tag data. The reference symbols can carry arbitrary data in the excitation signals. The full detailed process can be found in §II-D.

(c) How to improve energy efficiency for event-driven transmissions?

As backscatter communications are driven by carriers, the tag does not need to work in full speed all the time. Given limited harvested energy, we introduce a customized state machine to control how the tag works in different cases. The key of our state machine is a low-power listening state of which the power consumption is significantly lower than TX or RX states. The detailed design is described in §II-E.

To show the feasibility of our design, we prototype multiscatter using an FPGA and various commodity radios. Through extensive experiments, we show that

- Multiscatter achieves an average identification accuracy of more than 93% in the presence of 802.11b&n, Bluetooth, and ZigBee excitation signals with a sampling rate of 2.5 Msps. Specifically, the identification accuracies are 94.3% for 802.11n, 95.9% for 802.11b, 81.8% for BLE, and 99.9% for ZigBee.
- The maximal aggregate throughput of both productive and tag data is 278.4 kbps with a single Bluetooth radio, of which the productive data throughput is 141.6 kbps, and tag data throughput is 136.8 kbps. When the transmitter-to-tag distance is 0.2 m, the maximal communication for BLE is 71 m.

- With energy harvested from office light, it realizes backscatter rates of 12.4 pkts/s, 8.9 pkts/s, and 1.7 pkts/s with excitation signals of WiFi, BLE, and ZigBee, respectively.
- We also demonstrate that in the presence of various excitation signals, multiscatter can leverage such excitation diversity to provide uninterrupted communication and greater throughput gains, whereas single-protocol tag being idle when target signals are not available.

Contributions: We make the following contributions:

- We present a novel backscatter design that for the first time can effectively support multiprotocol identification, including WiFi, Bluetooth, ZigBee.
- We introduce overlay modulation, the first backscatter modulation that enables single-commodity-radio decoding by removing the dependency of data from the original channel. It is so flexible that various tradeoffs between tag-data rates and productive-data rates can be made for a range of practical applications.
- We introduce a customized state machine where a low-power listening scheme is employed for event-driven backscatter communication.
- We demonstrate a working system that is able to harness multiple excitation signals to provide much better connectivity in real scenarios. Empirically experiments confirm its feasibility and efficacy.

II. Multiscatter Design

We firstly give an overview of our multiscatter framework, then introduce how we reduce the listening power, obtain high-bandwidth baseband signals, correlate those signals to identify protocols, and modulate tag data onto excitation carriers.

A. Overview

As shown in Figure 2, the tag harvests RF power from abundant excitation signals in the 2.4 GHz ISM band and light from sunlight or indoor lamps. The system stays in low-power state when there is no excitation signal so that the listening energy can be reduced. When the carrier packet comes, the tag uses a high-bandwidth rectifier to acquire baseband amplitude signals and correlates sampled bits with pre-stored templates for identification. After the carrier is identified, it picks the corresponding modulation scheme to overlay tag data on top of productive carriers.

At a high level, this basic idea of multiscatter is simple. But there are several critical challenges to turn it into practice. First, the system should wake up from low-power listening state quickly enough when a carrier packet comes, because the carrier has a time duration as short as hundreds of microseconds. If the wakeup takes too much time, the tag will lose transmission chance. Second, the backscatter tag should avoid power-hungry components as much as possible, e,g, power amplifier. Also, it has very restricted resources for computation. For example, for baseband processing, we choose the FPGA that has the lowest power consumption on the market, Igloo nano AGLN250 [33]. Furthermore, decoding tag data should be easily done by a single commodity radio, e.g., Bluetooth on smartphones, for personal IoT applications.

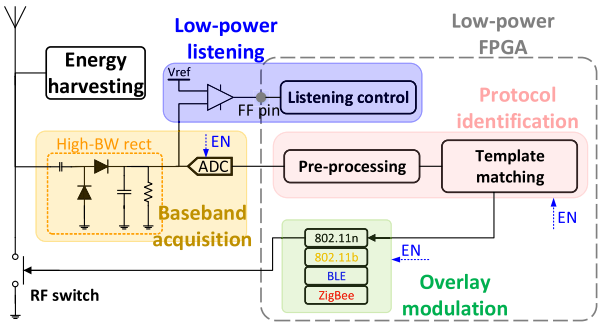


Fig. 2. Multiscatter overview. The tag firstly waits for carrier signals with low power consumption. When the excitation signal comes it obtains baseband signal through the high-bandwidth rectifier, then identifies the incoming carrier type, and finally modulates tag data onto the carrier.

B. High-Bandwidth Signal Acquisition

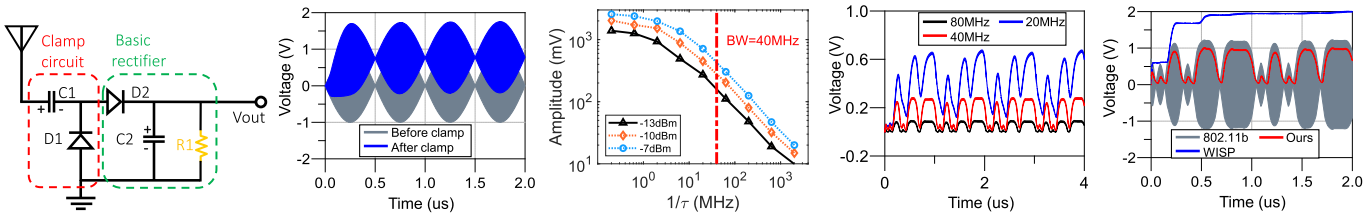
Packet detection is the first step of all wireless protocols, which indicates which type of packet is coming. For example, a typical 802.11b packet has a preamble of 144 μs long,[1] which is composed of 128 scrambled 1's and 16 Start Frame Delimiter (SFD) bits. For Bluetooth, it uses a preamble of 1 byte, defined as 0xAA, which is 8 μs.[2] As we need to identify those packets in the same band, the first question arises: how to obtain high-quality baseband signals for identification?

1) High-Bandwidth Envelope Detector: While active radios always use high-frequency mixers and low noise amplifiers to obtain baseband signals, backscatter tags do not have such luxury amenities due to energy constraints. We use a rectifier as shown in Figure 3a instead. The rectifier is sensitive to the signal energy, so it's able to extract the baseband amplitude information. We use the baseband amplitude for protocol identification. Still, a clamp circuit is added before the basic rectifier to improve the output amplitude.

As shown in Figure 3b, with input at 2.4 GHz, the clamp circuit effectively produces higher voltage. Here, someone may think of using a multi-stage rectifier to make even higher voltage; however, it not only reduces rectifying efficiency, but also distorts input signals very much. Thus it is mainly used for energy-harvesting purposes [34]. The second issue is the response rate of the rectifier. The quality of baseband signals from the rectifier is highly related to the discharging speed of the capacitor, which is related to the time constant of the RC circuit in Figure 3a: $\tau = R_1 C_2$. If it is too small or large, input signals would be distorted significantly. Suppose the carrier frequency of input RF signals is f_c and the baseband frequency is f_b, a proper τ should be chosen by $\frac{1}{f_c} \ll \tau \ll \frac{1}{f_b}$ [34]. In our case, $f_c = 2.4$ GHz and $f_b = 20$ MHz as bandwidths of Bluetooth and ZigBee are even lower. Between $\frac{1}{f_b}$ and $\frac{1}{f_c}$ there is still a large space to choose τ. We analyze

[1]The optional "short preamble" in 802.11b is 72 μs.

[2]As designed for low-power scenarios, BLE and Bluetooth are interchangeable in this paper.



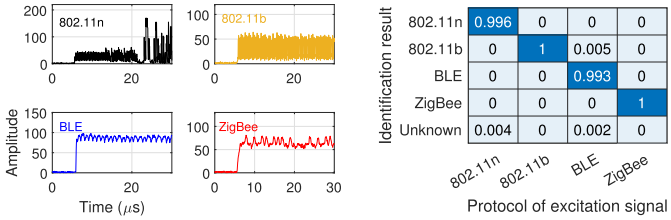
(a) We use a basic following a clamp to get baseband envelope. (b) Clamp voltage booster. (c) Amplitude with bandwidth. (d) Rectifier output waveform with different bandwidth. (e) Output waveform of our rectifier and WISP rectifier.

Fig. 3. With the help of optimized clamp and RC circuits, our rectifier is able to obtain high-bandwidth baseband signals from 2.4 GHz carriers.

the output amplitude of the rectifier when the $\frac{1}{\tau}$[3] is set to different values in the Agilent Advanced Design System. Figure 3c shows that the output amplitude drops quickly when $\frac{1}{\tau}$ increases. When the bandwidth $\frac{1}{\tau}$ increases 10 times the output amplitude will drop by about 70%∼80%. Larger $\frac{1}{\tau}$ ensures the similarity between output waveform and baseband amplitude signals, while smaller $\frac{1}{\tau}$ provides higher amplitude, that's a tradeoff. We compare rectifier output when the bandwidth is 20 MHz, 40 MHz, and 80 MHz. Results are shown in Figure 3d. As expected, the output amplitude is the highest with 20 MHz, but the signal waveform is distorted. With 80 MHz the condition is exactly the opposite. We choose $\frac{1}{\tau}$ to be 40 MHz. As shown in Figure 3e, the output fits incoming signals better and has decent signal strength compared to WISP [35].

To examine our rectifier's performance, we set the transmission power of 802.11n signals at 30 dBm,[4] the output voltage threshold of our rectifier at 0.07 V, and tag sensitivity at -18 dBm (typically -12∼-20 dBm for RFID tags); the achieved maximal downlink range is 1.8 m, which is less than the typical RFID reading range, ≈10 m. There are three main contributing factors for such a reduced downlink range. First, our rectifier has lower SNRs because it trades the output voltage (SNR) of the rectifier for fine-grained (high-frequency) baseband amplitude signals mainly due to the tuned resistor R_1. Second, our target signals at 2.4 GHz have shorter wavelengths (≈0.12 m) than RFID (≈0.33 m), which brings less than 15% of the received energy for an RFID tag along the same path. Third, we use a typical personal WiFi device that has an omni-directional antenna whereas an RFID reader is usually with directional patched antennas. Yet, for personal on-body sensors, 1.8 m downlink range is adequate to reuse excitation signals from smartwatches, cellphones, and laptops.

2) Template Matching: Our envelope baseband is different from the usual baseband after down conversion in active radios. Theoretically, frequency shift keying (FSK) signals (BLE) and phase shift keying (PSK) signals (802.11b and ZigBee) should have constant envelopes, which would render our baseband signals useless for identification. Fortunately, we observe that almost all modern wireless radios employ pulse-shaping at the transmitter for limiting bandwidth and reducing intersymbol interference. After pulse-shaping filters, sharp phase and frequency transitions are reduced, resulting in spectral efficient signals, which also makes constant envelopes not constant anymore. Still, the RF impairments including the IQ imbalance, the phase shift, and the frequency shift also contribute to the envelope waveforms. Those impairments are stable and could be used in identification.



(a) Acquired baseband signals (b) Identification accuracy

Fig. 4. We observe different envelope shapes for four different signals in (a) and achieve more than 99.3% identification accuracy in (b) for all four protocols when $L_t = 120, L_p = 40$.

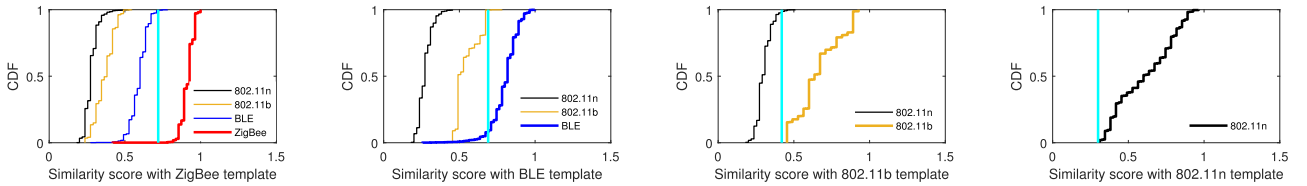
As shown in Figure 4a, all the baseband signals acquired manifest distinguishable envelopes. Next, we need to properly set templates and check the baseband quality. Specifically, we use an ADC to sample those baseband signals and correlate them with time-based templates, which measures how similar two vectors are. The correlation score is denoted as C. Assume the matching size is L_m. It should have two parts: a preprocessing window of size L_p and a template window of size L_t. The matching window is for correlation computation, the preprocessing window is for DC removal and normalization. How to set those parameters then? If we reuse the minimal length of packet detection fields for four protocols, the whole template window should be 8 μs, which is the length of the BLE preamble. And if the sampling rate is 20 Msps, then $L_t + L_p \leq L_m = 160$ samples. An exhaustive search shows that there are a number of combinations that can achieve more than 99% identification accuracy. For example, as shown in Figure 4 when $L_p = 40, L_t = 120$, the minimal identification among all four protocols is 99.3% and the average identification accuracy is 99.7%, demonstrating that the acquired baseband signals of four protocols are of high-quality for packet identification.

C. Low-Power Protocol Identification

Previously, we show that desirable identification accuracy can be achieved if computation resources are not a problem.

[3] $\frac{1}{\tau}$ expresses the bandwidth characteristics of rectifier output, and we treat it as a bandwidth parameter.

[4] 30 dBm power level is achieved by using a power amplifier.

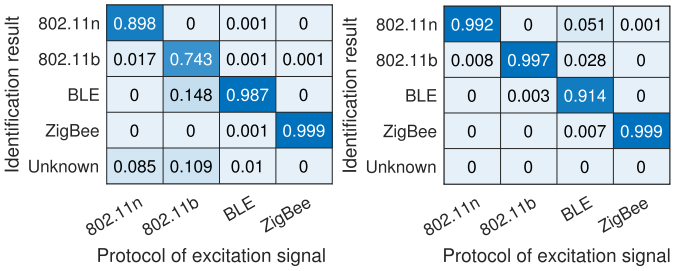


(a) Step 1: $C_{\text{Zigbee}} > Thrs_z$ (b) Step 2: $C_{\text{BLE}} > Thres_b$ (c) Step 3: $C_{802.11\text{b}} > Thrs_{11b}$ (d) Step 4: $C_{802.11\text{n}} > Thrs_{11n}$

Fig. 5. Our ordered matching, from ZigBee to BLE, to 802.11b, and to 802.11n.

TABLE II
COMPARISON OF DIFFERENT FPGA IMPLEMENTATIONS FOR MULTIPROTOCOL IDENTIFICATION

	Multipliers	Adders	D-Flip-Flops
802.11n[5]	120	119	33,341
802.11b	120	119	33,341
BLE	120	119	33,341
ZigBee	120	119	33,341
Total (Naive Impl.)	480	476	133,364
Nano FPGA Impl.			2,860



(a) Blind matching (b) Ordered matching

Fig. 6. Comparison of blind and ordered matching at a sampling rate of 10 Msps with quantization.

Our tag, however, uses an ultra-low-power FPGA, of which the power consumption is as low as 2 μW. In the following, we show how to fit the multiprotocol identification algorithm into this constrained FPGA.

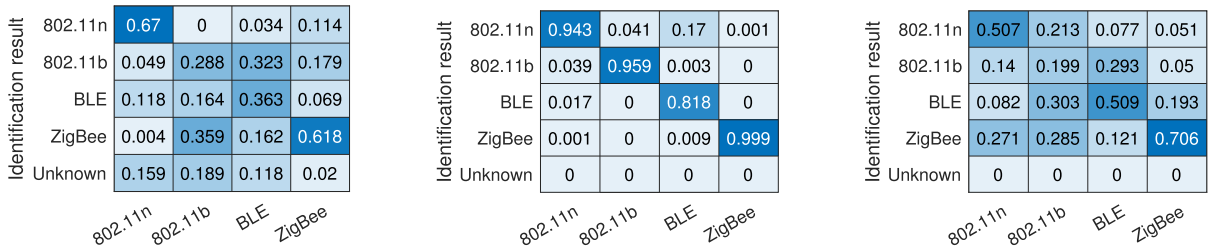
1) Low-Power Computation: To get a feel that why straightforward correlation-based matching is not feasible, we can estimate how many resources are needed for matching. For example, if the template size is 120, then we need 480 multipliers and 476 adders to do correlation on four templates. Since a 9*9 multiplier takes 259 D-Flip-Flops, and a 9*9 adder takes 19 D-Flip-Flops, the total D-Flip-Flops consumed would be 133,364 as shown in Table II. This is too many because an AGLN250 has only 6,144 D-Flip-Flops.

To address this, our solution is to do quantization and downsampling [36], [37], [38] at the same time. As quantization is a lossy process that reduces the precision of samples, we trade identification accuracy for meeting ultra-lower-power FPGA requirements. Specifically, we quantize each sample into 1 bit, which enables us to replace multipliers with adders. As a result, 2,860 D-Flip-Flops are enough to complete 4-protocol matching when the template size is 120. Then, we further employ downsampling to reduce the consumption of FPGA resources by downsizing the template. To check how quantization and downsampling affect detection accuracy, we show average accuracy results in Figure 6a. Compared to Figure 4b, we observe that quantization and downsampling do degrade detection accuracy but not too much.

2) Ordered Matching: During the above lossy process, we have another interesting observation: four excitation signals demonstrate noticeably different resilience. For example, as shown in 5a, more than 99% of ZigBee packets could be easily identified by setting a similarity threshold for ZigBee-template correlation, $C_{\text{Zigbee}} > Thres_z$, even when we downsample the baseband from 20 Msps to 10 Msps with quantization. Such a phenomenon motivates to use ordered matching, which makes decisions one after another, instead of blind matching that picks the highest score among the four. To obtain empirically optimized parameters for average identification accuracy, we perform brute-force search of all matching orders with discrete threshold values. It covers more than 200,000 traces of different ranges, scenarios, and protocols; the results are pretty much consistency and no location-sensitivity is observed. The ordered matching process is shown in Figure 5 and the corresponding accuracy results are demonstrated in Figure 6. We observe that the average identification accuracy increases from 0.906 for blind matching to 0.976 for ordered matching. Such performance gains should be attributed to different signal resilience because the four signals have so many differences, e.g., symbol size, modulation rate, and modulation scheme. For example, the baseband bandwidth of 802.11n, 802.11b, ZigBee, and BLE are 20 MHz, 11 MHz, 2 MHz, and 1 MHz, respectively. With the tag sampling rate lower than 10 Msps, the sampled 802.11n baseband amplitude will encounter aliasing. So the identification of 802.11b/n will be seriously influenced. But for ZigBee and BLE, 10 Msps is high enough, and the corresponding identification keeps high accuracy. Firstly identifying BLE and ZigBee, and then identifying 802.11b/n will help improve the accuracy.

With the help of ordered matching, we attempt to keep reducing sampling rates and find that when the sampling rate is 2.5 Msps, it becomes tough to differentiate the four signals. The average identification accuracy is only 0.485 as shown in Figure 7a. Therefore, we intend to prolong the matching window, i.e., finding the maximal matching window size. We observe that only BLE and 802.11n are the limiting factors

[5]The template size is 120 and each samples takes 9 bits.



(a) 8μs-matching window at 2.5 Msps (b) 40μs-matching window at 2.5 Msps (c) 40μs-matching window at 1 Msps

Fig. 7. Using an extended matching window of 40 μs, the average identification accuracy improves from 0.485 in (a) to 0.93 in (b). Nevertheless, if we continued to reduce the sampling rate to 1 Msps, the accuracy is not desirable.

since the preambles of ZigBee and 802.11b are longer than 100 μs. For BLE packets, the access address of advertising packets stays the same, which means we can extend the matching window size to 40 μs by including this broadcasting address. Meanwhile, for 802.11n, behind the legacy preamble, there are HT-STF and HT-LTF fields designed for MIMO support, which are more than 20 μs. Hence, our extended matching window size can be safely set at 40 μs for all the four protocols. Through such an extension, the average identification accuracy at 2.5 Msps is boosted from 0.485 to 0.93 as shown in Figure 7b. Empirically, we set the lowest sampling rate at 2.5 Msps if applications demand high accuracy (> 0.9) because 1 Msps can only provide an average identification accuracy of 0.5 as shown in Figure 7c.

A few points are worth noting:

1) Although our ultra-low-power FPGA supports a limited storage space of 36 kb for both code and data, the storage overhead of four templates with the extended length is 400 bits, which only costs 1.1% of the total storage space.
2) Due to the effects of analog random noise and quantization noise, we optimize the ADC performance by tuning the reference voltage to match the full-scale range of the input signal because more of the output codes are used with the smaller range of input voltages.

D. Overlay Modulation

After excitation signals are identified, the next important task is how to embed tag data onto those carriers. We first show why state-of-the-art systems are difficult to fit in with personal radios, then propose our novel overlay modulation scheme, and summarize its pros and cons.

1) Motivation: Being able to handle productive-data carriers is an important feature for backscatter, as neither dedicated RFID readers nor single-tone generators [22] are commonly available for personal IoT sensors. The key enabler of it is the codeword translation, which encodes tag data by changing a valid codeword into another. Yet, those state-of-the-art systems [10], [27], [30] share two major drawbacks. First, the decoding quality of tag-data is highly dependent on the data from the original channel. In other words, when the original channel becomes unstable due to occlusion or mobility, it is difficult to decode tag data even when the data from the backscattered channel is error-free. Second, large modulation offsets make two-receiver synchronization necessary because tag-data decoding requires to XOR two codewords of the same index from two receivers. To avoid synchronization overhead, PLoRa makes use of a USRP that covers a wide band, so it samples the original and backscatter channels at the same time. Apparently, neither synchronizing two receivers nor requiring extra specialized hardware is favorable for personal IoT sensors because single personal radios, e.g., WiFi, Bluetooth, are more typical and popular.

2) Reference-Based Tag Modulation: To make backscatter work with productive carriers and single commodity radios at the same time, we novelly propose overlay modulation, which is to modulate tag data on top of modulated (productive) carriers. This idea is made based on an important observation: *codeword translation can be realized in a single data stream*, instead of involving data from two channels in previous systems, which for the first time completely removes the dependency of data from the original channel. We name it reference-based overlay modulation as it is inspired by both pilot symbols widely used in wireless communication [39] and the overlay network that is built on top of another network [40]. The detailed workflow is as follows. As shown in Figure 8, in overlay modulation, a productive carrier consists of several modulatable sequences. Each modulatable sequence is κ-symbol long. The first symbol is the reference symbol that carries productive data, and the rest $\kappa-1$ symbols have exactly the same content as the reference symbol and are modulatable for tag data. To generate such carriers, it only needs to spread the original symbol for κ times, so we call κ the spread factor for productive data. Note that the main usage of reference symbols is to demodulate tag data, and it is in the payload part, so it would not affect channel estimation or signal acquisition.

Upon receiving productive carriers, the tag first applies reference-symbol demodulation to obtain productive data and then goes through a reverse codeword translation to demodulate tag data. The real beauty of overlay modulation is that the decoding both productive and tag data happens on a single packet. As there are two kinds of modulation involved in overlay modulation, reference-symbol modulation, which comes from original carriers, and tag-data modulation, which adopts codeword translation from Hitchhike [29] and FreeRider [30],

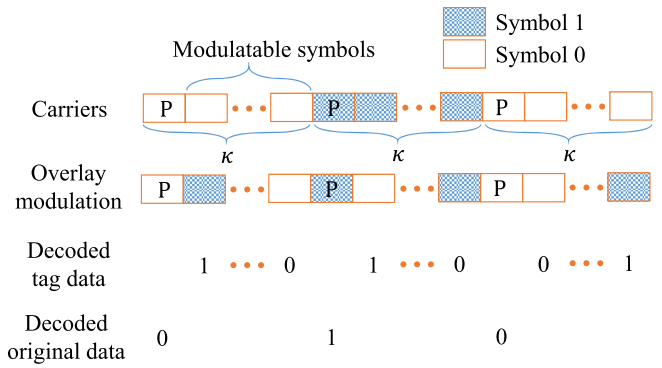


Fig. 8. Overlay modulation where the carrier is composed of a couple of modulatable sequences. Each sequence has a reference symbol carrying productive data and modulatable symbols for tag data.

we will discuss how to do tag-data modulation and demodulation with WiFi,[6] ZigBee, and Bluetooth as follows.

802.11b. For 802.11b excitation signals, reference symbols support DSSS-BPSK, DSSS-DQPSK, and CCK modulation (same as original 802.11b modulation). Despite various modulation schemes for reference symbols, we observe that BPSK-based tag-data modulation is compatible with all of them. Specifically, if the excitation signal is identified as 802.11b, we first frequency shift it to another channel and thus avoid creating interference in the original channel [22], [25]. Then, the tag modulates each tag bit by simply shifting phase 0 or π. For example, to modulate a tag bit 1, we can phase shift an 802.11b symbol for π, and to modulate a tag bit 0, we keep the phase unchanged. To demodulate tag data, a simple XOR operation of the reference symbol and the modulated tag symbol is adequate.

Ideally, a tag bit can be modulated onto one 802.11b symbol. Unfortunately, due to backscatter signal and modulation errors, the decoding performance becomes unstable. Inspired by DSSS in ZigBee that uses long modulation length to combat low SNRs [41], we define γ, the spreading factor for tag data, which means using γ symbols to modulate one tag bit. For example, if $\gamma = 8$, it means a reference symbol (same as data symbol) takes 8 μs for 1 Mbps 802.11b. Nevertheless, the received bitstream from commodity 802.11b radios may contain reference symbols that are not all 0s or 1s. To address this, we introduce majority voting to decode reference symbols.

802.11n. For 802.11n signals, the situation becomes a bit more complicated as 802.11n involves OFDM. Reference-symbol modulation for 802.11n includes OFDM-BPSK, OFDM-QPSK, and OFDM-QAM. We observe that compared to 802.11b, even OFDM incorporates multiple orthogonal subcarriers, its main operation, IFFT, is still a linear operation [42], i.e., BPSK-based tag-data modulation stands. The difference is the unit of tag-data modulation becomes OFDM symbol, 4 μs for each. Another thing is that as the scrambler and BCC encoder are not completely compatible with codeword translation [30], which may lead to broken structures, tag-data modulation cannot turn an OFDM-symbol of all 1s into an OFDM-symbol of all 0s. The solution is to apply majority voting for the middle half part of modulated symbols [10].

ZigBee. Reference-symbol modulation for ZigBee adopts offset quadrature phase-shift keying (OQPSK) [41]. In particular, each ZigBee symbol has 4 bits, which are mapped into a PN code of 32 chips. The chips are reorganized into IQ series where there is constant half a chip offset in-between. While such offset is designed to reduce PARP, it presents challenges for BPSK-based tag-data modulation because a phase shift of π would damage this half-a-chip offset structure. The solution is to increase γ. This way, the first modulated ZigBee symbol may be incorrect, but the rest symbols can be decoded successfully because commodity ZigBee radios pick the best-matched sequence among 16 predefined PN sequences. According to our experiments, $\gamma = 3$ can achieve BERs around 0.1%.

Bluetooth. If we identify the carrier as Bluetooth, it would employ FSK-based tag-data modulation, instead of PSK for the previous three kinds of signals. According to the specification [43], reference-symbol modulation for Bluetooth should adopt Gaussian Frequency-Shift Keying (GFSK): f_0 for symbol 0 and f_1 for symbol 1. For example, commodity BLE radios have a modulation index of 0.5, which is $\frac{f_1 - f_0}{f_m}$, where f_m is the modulation frequency. If the modulation frequency is 1 MHz, then $f_1 - f_0 = 500$ kHz. Accordingly, our tag-data modulation can encode a bit 1 by shifting a frequency of $\Delta f = 500$ KHz, which turns a bit 1 to a bit 0, and there is no frequency shift if we need to modulate a tag bit 0.

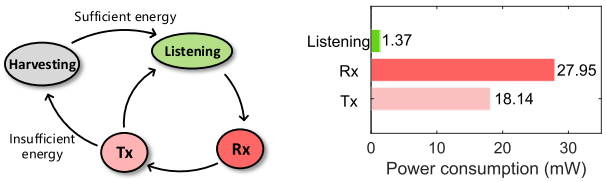
3) Choice of Proper Spreading Factor γ: The spreading factor γ is a crucial parameter for system performance, we now consider how to choose it. Smaller γ means fewer carrier symbols are used for a tag bit. This will cause higher throughput and higher BER, and vice versa. But higher throughput and lower BER are wanted. Optimizing BER will cause larger γ, while optimizing leads to smaller γ. How to make a tradeoff and get proper γ?

We use the quantity of information [44], [45] to combine BER and throughput. Let X be a discrete random variable, and its possible states are $x_1, x_2, \ldots$. The information entropy of X is defined to be $H(X) = -\sum_i p(x_i) log_2 p(x_i)$ where $p(x_i)$ means the probability of $X = x_i$. After receiving a message Y, the information entropy of X becomes: $H(X|Y) = -\sum_i p(x_i|Y) log_2 p(x_i|Y)$, where $p(x_i|Y)$ is the conditional probability of $X = x_i$. The quantity of information about X contained in Y is: $I(X,Y) = H(X) - H(X|Y)$. In our case, X is a single bit with states of '0' and '1'. Initially, $p(0) = p(1) = \frac{1}{2}$. Y is one decoded tag bit in the receiver, and it can be '0' or '1'. Let $B(\gamma)$ denote the BER.[7] After Y is recovered, $p(0|1) = p(1|0) = B(\gamma)$, and $p(1|1) = p(0|0) = 1 - B(\gamma)$. So the information contained in a tag bit is: $I(\gamma) = I(X|Y) = 1 + B(\gamma) \times log_2 B(\gamma) + [1 - B(\gamma)] \times log_2 [1 - B(\gamma)]$.

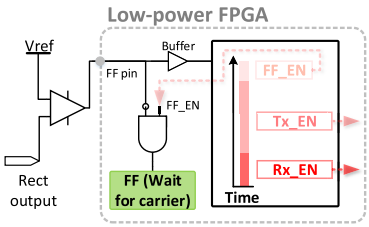
The backscatter throughput is $\frac{1}{\gamma \times T_S}$, where T_S is the duration of a carrier symbol. For 802.11b/n, ZigBee, and BLE,

[6]Currently, we mainly focus on two types of WiFi: (1)DSSS and CCK modulation: 802.11b, and (2)the OFDM modulation that covers 802.11a/g/n/ac/ax

[7]BER is a function of γ.



(a) Four system states. (b) Power at different states.



(c) Low-power listening control logic.

Fig. 9. Multiscatter low-power listening design.

T_S is 1 μs, 4 μs, 16 μs, 1 μs, respectively. We define the tag information rate: $InfoRate = I(\gamma) \times \frac{1}{\gamma \times T_S}$ to combine both BER and throughput. It can be seen as the tag transmission capacity in some way. Equal tag throughput without error can be realized if a proper coding scheme is used on tag data. We can choose γ to optimize the $InfoRate$ instead of BER or throughput.

4) Summary: While the way of modulating symbols is inspired by codeword translation [29], [30], our overlay modulation is built on top of it and beyond. The major differences are as follows:

- Overlay modulation is the first to enable productive and tag data co-existence in the same packet, resulting in that only a single commodity radio is adequate for decoding.
- The spectral efficiency is largely improved as the required decoding spectrum is the same as the original channel, whereas prior work [10], [27], [29], [30], [31] demands twice of that.
- Introducing reference symbols brings two limitations. First, ambient signals cannot be excitation carriers for multiscatter tags. Second, it reduces the throughput of tag data. Yet, various tradeoffs can be made between the productive and tag data throughputs by simply adjusting κ, which can be as short as 2, and as long as the full payload. In short, overlay modulation sacrifices the freedom of arbitrary productive data for simpler decoding of tag data.

E. Event-Driven State Machine

We have shown that protocol identification and overlay modulation can be realized if the power is sufficient and stable. But the peak power consumption of multiscatter prototype is too high for a battery-free system. To reduce system power, we have two considerations:

(1) FPGA: The AGLN250 FPGA consumes two-thirds of total power. It has a "Flash*Freeze" low-power mode in which the FPGA core is static but with all register states kept. We can use this to reduce FPGA active time for power reduction.

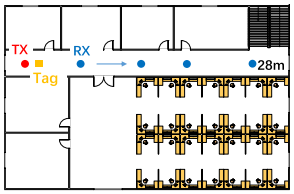
(2) Multiscatter prototype: Multiscatter system can be kept in low-power mode when there is no excitation signal, too. For this, we design four working states for multiscatter as:

- *Harvesting*. Initially, there is no energy and the system stays in the *Harvesting* state until sufficient energy has been stored. In this state, only the harvester runs normally and other parts are shut down.
- *Listening*. With harvested energy, the system firstly waits for carrier packets. Power-consuming operations such as sampling baseband amplitude signals, identifying excitation, and modulation are all stopped. Only the comparator and oscillator are kept active.
- *Rx*. Multiscatter is designed to stay at this state only when an excitation comes but has not been identified. In this state, the system runs at full speed to identify the excitation protocols and conduct frequency-shifting. All components are in active mode and the system power consumption is the highest.
- *Tx*. In this state, the system reads sensor data and conducts overlay modulation for sensor data transmission. The ADC can be shut down as identification has been realized. We bound the sensor reading procedure with backscatter transmission to reduce the active time for power reduction.

It can be easily seen that in 3 working states *Rx* has the highest power consumption, and *Listening* has the lowest. As Figure 9b shows, *Rx* consumes 27.95 mW, *Tx* consumes 18.14 mW, and the *Listening* state needs 1.37 mW. The state switching is shown in Figure 9c. The switching control is realized by three enable signals: *Rx_EN*, *Tx_EN*, and *FF_EN*. They push multiscatter to *Rx*, *Tx*, *Listening* states, respectively. The specific procedures are as follows:

- *Entering and exiting Harvesting state.* After enough energy has been harvested, the system power supply is restored. The "Flash*Freeze" control through *FF pin* is initially enabled, and the FF pin is logic low, the system enters *Listening* state. If stored energy is not enough for the next round of sensor data reading and transmission, the harvesting management chip cuts off the power supply and the system enters *Harvesting* state.
- *From listening to Rx.* When excitation comes, the rectifier output exceeds the reference voltage of the comparator, and the FF pin becomes logic high. Then the FPGA wakes up and sets $Rx\_EN = 1, Tx\_EN = 0$, and $FF\_EN = 0$. The system enters *Rx* state. The rectifier causes a time delay within 0.5 μs, the comparator causes 1 μs, and the FPGA needs about 1 μs to wake up. So the transition delay is 2.5 μs, it's is short enough compared to the template duration of 40 μs. The oscillator is kept active at *Listening* state because it needs as long as 200 μs to stabilize after power recovery.
- *From Rx to Tx.* As long as the protocol-identification has been realized, the enable signals are: $Rx\_EN = 0, Tx\_EN = 1$, and $FF\_EN = 0$. The ADC sampling and identification program are disabled, while sensor reading and overlay modulation are enabled, and multiscatter enters the *Tx* state.

(a) Multiscatter tag prototype.



(b) Experiment deployment.

Fig. 10. Our tag prototype size is similar to credit card, around $60/each, and the experimental area is 30m*50m.

- *From Tx to Listening*. After backscatter transmission has been finished, the enable signals become to: $Rx\_EN = 0, Tx\_EN = 0$, and $FF\_EN = 1$. The tasks in state *Tx* and *Rx* are disabled, and the "Flash*Freeze" control by *FF pin* is enabled. When the carrier packet finishes, the *FF pin* becomes logic low and multiscatter enters the *Listening* state.

III. Implementation

We build a prototype of multiscatter using various commodity radios and a ultra-low-power FPGA. The implementation is detailed as follows.

A. Off-the-Shelf Prototype

Our prototype consists of three main parts: an RF front-end, an FPGA for baseband processing, and a power harvesting block. As shown in Figure 10a, the prototype has a similar size with a credit card.[8] There are two antennas in the front end. One is connected to the rectifier for system wakeup and protocol identification. The rectifier circuit is composed of diodes, resistors, and capacitors. The comparator NCS2200 compares the envelope amplitude with a reference. When carrier signals come, the comparator output is high and the system will wake up from low-power listening. An LTC2366 ADC which runs at 2.5 Msps is used to sample baseband amplitude for identification. The other antenna is connected to an ADG902 RF-switch. The system uses the TI harvesting chip BQ25570 to manage harvested energy from the solar panel MP3-37 when RF power is not enough. The harvested energy is stored in a 1000 μF capacitor. All the baseband processing, including multiprotocol identification, phase and frequency modulation are realized in low-power AGLN250 FPGA.

B. Protocol-Identification Power Efficiency

To examine how much power savings our multiprotocol identification design achieves, we compare it against other variant implementations without quantization or downsampling. The competition metric is the simulated power consumption on a XILINX Artix-7 FPGA because variant implementations are too complex to deploy on an AGLN250. Results are shown in Table III.[9] At 20 Msps sampling rate, quantization reduces power consumption from 564 mW to 12 mW. Further, with 2.5 Msps sampling rate and quantization, the consumed power drops to 2 mW, which translates to a 282× power reduction.

[8]For ease of current testing we set a testing port for every active component.

[9]LUT (Look-Up-Table) is the basic configurable logic element in an FPGA. Usually, the number of LUTs used is proportional to the IC-simulation power consumption.

TABLE III

Required Hardware Resources and Power Consumption of Protocol-Identification Algorithms

Setup	Power(mW)	LUTs
20MS/s, no ±1 quan.	564 (100%)	34751
20MS/s, ±1 quan.	12 (2.1%)	1574
2.5MS/s, ±1 quan.	2 (0.35%)	1070

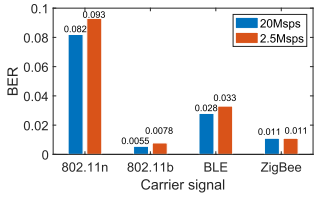


Fig. 11. BER rises slightly after down-sampling.

TABLE IV

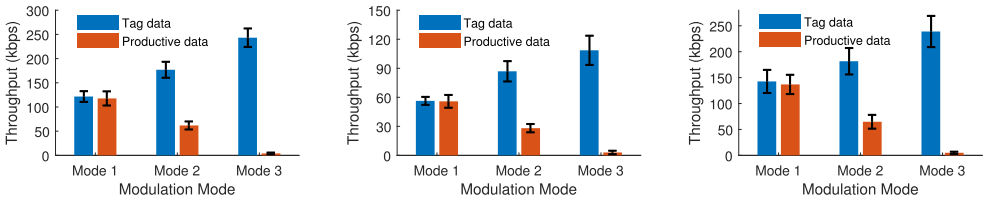
Comparison of PCB and IC Power Consumption

Implementation	Component power consumption					Total
	Digital core	ADC	Oscillator	Switch	Comparator	
Prototype	18.19 mW	7.89 mW	0.59 mW	1.1 mW	0.18 mW	27.95 mW
IC	275 μW	237 μW	202 μW	4 μW	10 μW	728 μW

C. Prototype and IC-Implementation Power Consumption

We use a Keysight 34450A digital multimeter to measure the currents of prototype components to get their power consumptions. The result is shown in Table IV. In the *Rx* state, multiscatter consumes 27.95 mW, including 18.19 mW by the FPGA, 7.89 mW by the ADC, 0.59 mW by the oscillator, 1.1 mW by the RF-switch, and 0.1 8mW by the comparator. The FPGA consumes over two-thirds of total power. That's because the FPGA not only controls the system, but also generates the 30 MHz clock signal using the phase-locked loop (PLL) and digitally processes it for frequency-shifting.

To optimize the power consumption, we present an IC implementation simulation as done in many related works [27], [29]. As the PLL consumes lots of power, we simulate a ring-oscillator to directly generate the 30 MHz signal. The FPGA only realizes the digital core. Its power is estimated to be 275 μW in the Libero power verification tool. The ADC design is out of our scope, so we take the power consumption from reference works for power estimation. [46], [47] show that the ADC with more than 5 Msps consumes less than 237 μW. We use a 2.5 Msps ADC, and we also estimate the ADC power to be 237 μW. The analog components are simulated using the Cadence IC6.17 Virtuoso software and the TSMC 0.18μm CMOS process design kits. The ring-oscillator consumes about 202 μW, and the total power is reduced to 728 μW.



(a) WiFi 802.11b (b) WiFi 802.11n (c) BLE

Fig. 12. Tradeoffs between productive data and tag data throughputs under different modes.

The ADC consumes a large portion of the total power. Further reducing the ADC sampling rate helps realize better power efficiency. Adjusting the rectifier bandwidth and designing new identification algorithms besides the template matching may enable us to further reduce the sampling rate. Lower rectifier bandwidth means less spectrum aliasing with further down-sampling. And the phenomenon that envelope waveforms of BLE and ZigBee are closer to constant ones can help identify WiFi from BLE and ZigBee, making the identification task easier. Then lower sampling rate may be achievable. We will try this in future work.

But we should also notice that 728 μW is still much lower than low-power active radios such as ZigBee. We tested two commodity devices, the Microchip ATmega256RFR2 and the TI CC2530F256. The former consumes 54.12 mW and the latter consumes 113.19 mW, which are over $50\times$ higher than multiscatter tag. The low-power of those radios can be realized by pushing the system into sleep mode, in which the power can be reduced to several microwatts, as much as possible. This scheme is similar to our low-power listening.

D. Experimental Setup

Figure 10b shows the floor plan. All devices are placed in the hallway; we deploy a multiscatter tag 0.8 m away from the exciter, then we move the receiver away from the tag and measure the backscatter performance.

For WiFi, we use Qualcomm Atheros AR938X NICs as both productive carrier generator and receiver, and set the transmission rate at 1 Mbps for 802.11b and MCS=0 for 802.11n. For BLE, we employ a TI CC2540 radio as the transmitter at 1 Mbps and a TI CC2650 as the receiver. We empirically confirm that the maximum advertising packet rate is stable around 70 packets/s. For ZigBee, we adopt a TI CC2530 radio as the transmitter and a TI CC2650 radio as the receiver. The maximal packet rate for CC2530 is about 20 packets/s. As our overlay modulation requires to obtain raw data bits on the physical layer, the CRC (cyclic redundancy check) functions of NICs are turned off in our experiments.

IV. Evaluation

A. End-to-End Performance

1) Synchronization With Down-Sampling: Besides identification, synchronization is another point to consider after down-sampling. The peak of template matching may appear at an unexpected time, so the energy level instead of the template-matching peak is used for synchronization when the sampling rate is lower than 10Msps. As it's non-trivial to directly measure the synchronization accuracy, we conduct end-to-end experiments to show that energy-based synchronization after down-sampling is enough for overlay modulation. We compare the tag data BER with template-matching based synchronization at 20Msps and that with energy-based synchronization at 2.5Msps. The result is shown in Figure 11. As can be seen, after down-sampling, the BER is only slightly influenced. So the following experiments are all conducted with a sampling rate of 2.5Msps and energy-based synchronization.

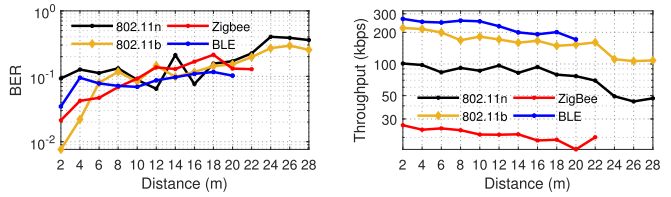
TABLE V
THREE MODES THAT CARRY DIFFERENT AMOUNT OF PRODUCTIVE DATA AND TAG DATA BY ADJUSTING κ

	Mode 1 κ	Mode 2 κ	Mode 3 κ
802.11b, $\gamma = 4$	8	16	$4n$ [10]
802.11n, $\gamma = 2$	4	8	$2n$
BLE, $\gamma = 4$	8	16	$4n$
ZigBee, $\gamma = 2$	4	8	$2n$

2) Tradeoffs Between Productive and Tag Data: According to the design of overlay modulation, γ determines how long a reference symbol is and κ defines the number of modulatable symbols. Since the reference symbol carries productive data and modulatable symbols carry tag data, we can adjust ratios of the two to make tradeoffs. In particular, we define three modes as shown in Table V. In mode 1, the number of reference symbols is the same as that of modulatable symbols, which would make throughputs of the two pretty close. Compared to mode 1, mode 2 increases the ratio of modulatable symbols to reference symbols from 1:1 to 3:1. Mode 3 pushes this to an extreme, which allows modulatable symbols to be as many as possible and only a single bit of productive data would be transmitted.

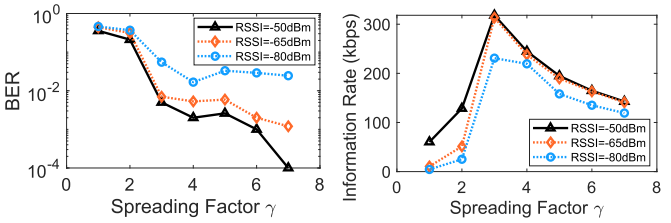
The throughputs with three modes are shown in Figure 12. We observe that in mode 1, the achieved productive and tag data throughputs are roughly the same across different excitation signals. The maximal aggregated throughput is 278.4 kbps for BLE, of which the productive data throughput is 141.6 kbps, and tag data throughput is 136.8 kbps. For mode 2, the tag data throughput surges because the number of modulatable symbols is $3\times$ than that of reference symbols. In mode 3, as expected, we barely see any throughput for productive data. In contrast, tag data throughput is maximized.

[10] $n = \lfloor \frac{I}{\gamma} \rfloor$ and this formula applies to all ns in the table.



(a) BER (b) Throughput

Fig. 13. Backscatter BER and throughput across distances in LoS deployment.



(a) BER with γ. (b) Information rate with γ.

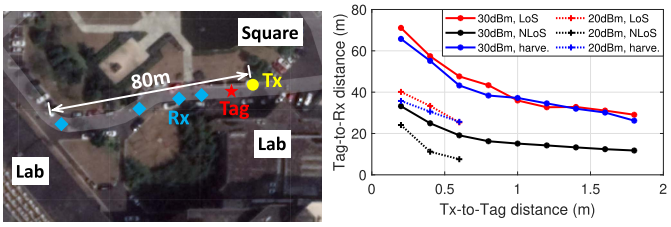
Fig. 14. BER drops when γ increases. Multiscatter achieves the highest information rate when $\gamma = 3$.

According to different application requirements, the tradeoff between productive data and tag data can be simply made by choosing a proper κ. Since mode 1 provides the best balance between two kinds of data, mode 1 is our default setting in the rest of the evaluation.

3) Backscatter Transmission Performance: Next, we intend to examine the backscatter transmission performance. The Tx-to-tag distance is set to 0.8 m. We measure the BER and throughput of multiscatter. Figure 13 shows that the maximum backscatter communication range of WiFi(802.11b/n), ZigBee, and Bluetooth are 28 m, 22 m, and 20 m, respectively. From Figure 13a, we can see that four protocols can still maintain low BERs when the tag is as far as 16 m away from the receiver. Figure 13b demonstrates that multiscatter achieves maximal aggregate throughputs of 278.4 kbps, 219.8 kbps, 101.2 kbps, 26.2 kbps for Bluetooth, 802.11b, 802.11n, and ZigBee, respectively.

4) Tuning Spreading Factor γ: To find proper γ, we set γ a positive integer and RSSI to about -50dBm, -65dBm, and -80dBm. The excitation is BLE packets. Results are shown in Figure 14a. BER drops quickly with the increase of γ. When $\gamma = 1$, the BER reaches even 46%. When γ exceeds 3, the BER is below 1%. The effective information rate is shown in Figure 14b. We can see that the backscatter information transmission efficiency is highest when the spreading factor $\gamma = 3$, so we can choose γ to be 3.

5) Impact of Tx-to-Tag Distance: Communication range is also very important [48], [49], [50], [51]. We conduct an experiment to investigate the influence of the Tx-to-tag distance on the communication range. This experiment is conducted on a road next to a Lab and a square, as shown in Figure 15a. A USRP N210 with daughterboard SBX-40 is used to generate BLE packets of different power levels. Then a power amplifier is used to boost those power levels to 20 dBm and 30 dBm. We set different Tx-to-tag distances



(a) The experiment scenario. (b) Communication range.

Fig. 15. Communication range with Tx-to-tag distance.

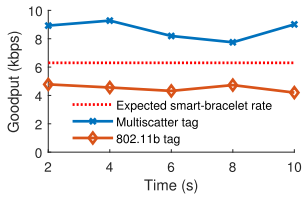


Fig. 16. Multiscatter intelligently picks carrier signal.

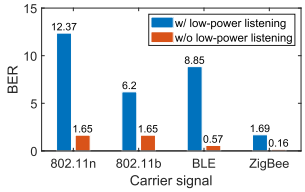


Fig. 17. The tag packet rate rises significantly with low-power listening.

and observe the corresponding achievable tag-to-Rx distances. Both the LoS and NLoS deployments are tested. A steel plate is placed between the tag and the Rx in NLoS deployment. Still, we also test the LoS communication range with harvested energy.

The result is shown in Figure 15b. With the Tx power of 30 dBm, the achievable Tx-to-tag distance is about 1.8 m, beyond which the tag will fail to detect excitation. While with 20 dBm Tx power, it's 0.6 m. With LoS deployment and Tx power of 30 dBm, the maximum communication range is 71 m when the Tx-to-tag distance is 0.2 m. And when the Tx-to-tag distance is increased to 1.8m, the communication will not exceed 29 m. With Tx power of 20 dBm, the maximum communication is limited below 40 m. In NLoS deployment, the achievable communication range is no more than half of that in LoS deployment. While using harvested energy, both the achievable Tx-to-tag distance and the maximum communication range encounter almost no decrease.

B. Leveraging Excitation Diversity

In this experiment, a smart bracelet has to deliver a goodput of more than 6.3 kbps for on-body monitoring, and there are abundant 802.11n and few 802.11b excitation signals. As shown in Figure 16, after evaluating the excitation rates of all signals, multiscatter tag detects that the current 802.11n excitation is with the highest backscattered goodput. Thus, it intelligently selects 802.11n as its source to accomplish the

goodput goal. In contrast, the 802.11b tag fails to meet the requirement because 802.11b excitation signals are spotty.

C. Low-Power Listening

To show the effectiveness of the low-listening power scheme, we compare the packet rates with and without it when 500 Lux office light is the energy source. Results are shown in Figure 17. The effective backscatter packet rate is improved for about $7.5\times$, $3.7\times$, $15.3\times$, and $10.2\times$ when the carrier signal is 802.11n, 802.11b, BLE, and ZigBee, respectively.

V. Related Work

For the last decade, turning backscatter into general-purpose communication for IoT networks has been a hot topic. Frequency-shifting is proven to improve SNRs in backscatter deployment [25], [52]. BackFi [24] improves backscatter throughput to high data rates of 5$\sim$300 Mbps. Despite its high throughput, the required full-duplex radios for self-interference cancellation are hard to realize for off-the-shelf devices. Passive WiFi [26] is the first backscatter design that decouples low-power digital baseband processing with power-consuming carrier generation and achieves up to 11 Mbps data rate. The key enabler of this approach is a dedicated plug-in device that transmits single tones out of the WiFi bands, which is also used in LoRa backscatter [28] and BLE-backscatter. To overcome the limitation of infrastructure support, interscatter [22] comes in. It novelly turns a Bluetooth device into a single-tone generator using reserve-whitening techniques. While everything seems perfect, the severe problem is reserve-whitening forbids using productive signals as carriers.

Hitchhike [29] is the first work that enables productive backscatter with commodity devices at the symbol level. Enabling such productive backscatter significantly widens backscatter sources and makes ubiquitous backscatter vision closer. FreeRider [30], X-Tandem [31], PLoRa [27] expand this idea from different perspectives. But the common fundamental issue is that the decoding process requires the productive data in the original channel and the tag-modulated data in the frequency-shifted channel.

Along this research line, multiscatter is inspired by and built upon all the aforementioned efforts and makes two fundamental differences. First, it greatly broadens backscatter sources by supporting multiprotocol identification, including WiFi, Bluetooth, and ZigBee in the most crowded 2.4 GHz ISM band. For the first time, the backscatter design is not restricted to only one kind of carrier as in prior work. Second, it encodes tag data on top of productive data and can be decoded using a single commodity device, removing the barrier to fast adoption with smart devices.

VI. Conclusion

We have presented multiscatter, a novel backscatter design that can identify multiple excitation signals and take productive carriers for backscatter. We have built the hardware prototype and conducted extensive experiments to verify the feasibility and efficacy. We believe that supporting multiple excitation signals is a significant step towards general-purpose battery-free communication for IoT, since it can be seamlessly incorporated into widely deployed wireless infrastructure.

References

[1] W. Gong, L. Yuan, Q. Wang, and J. Zhao, "Multiprotocol backscatter for personal IoT sensors," in *Proc. 16th Int. Conf. Emerg. Netw. EXperiments Technol.*, Nov. 2020, pp. 261–273.

[2] A. Varshney, C. Pérez-Penichet, C. Rohner, and T. Voigt, "LoRea: A backscatter architecture that achieves a long communication range," in *Proc. 15th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2017, pp. 1–14.

[3] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low-power backscatter networks," in *Proc. ACM SIGCOMM*, 2012, pp. 61–72.

[4] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, "Empowering low-power wide area networks in urban settings," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 309–321.

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

[7] D. Vasisht, G. Zhang, O. Abari, H.-M. Lu, J. Flanz, and D. Katabi, "In-body backscatter communication and localization," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2018, pp. 132–146.

[8] Y. Ma, Z. Luo, C. Steiger, G. Traverso, and F. Adib, "Enabling deep-tissue networking for miniature medical devices," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2018, pp. 417–431.

[9] J. Jang and F. Adib, "Underwater backscatter networking," in *Proc. ACM Special Interest Group Data Commun.*, Aug. 2019, pp. 187–199.

[10] J. Zhao, W. Gong, and J. Liu, "Spatial stream backscatter using commodity WiFi," in *Proc. 16th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2018, pp. 191–203.

[11] 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.

[12] 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 Conf.*, Aug. 2016, pp. 384–397.

[13] M. Rostami, J. Gummeson, A. Kiaghadi, and D. Ganesan, "Polymorphic radios: A new design paradigm for ultra-low power communication," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2018, pp. 446–460.

[14] J. Zhao, W. Gong, and J. Liu, "Towards scalable backscatter sensor mesh with decodable relay and distributed excitation," in *Proc. 18th Int. Conf. Mobile Syst., Appl., Services*, Jun. 2020, pp. 67–79.

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

[16] S. Chen, W. Gong, J. Zhao, and J. Liu, "High-throughput and robust rate adaptation for backscatter networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2120–2131, Oct. 2020.

[17] W. Gong, J. Liu, K. Liu, and Y. Liu, "Toward more rigorous and practical cardinality estimation for large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1347–1358, Jun. 2016.

[18] W. Gong et al., "Channel-aware rate adaptation for backscatter networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 751–764, Apr. 2018.

[19] W. Gong, S. Chen, and J. Liu, "Towards higher throughput rate adaptation for backscatter networks," in *Proc. IEEE 25th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2017, pp. 1–10.

[20] 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.

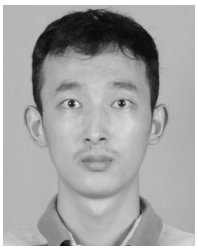
[21] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith, "Ambient backscatter: Wireless communication out of thin air," in *Proc. ACM SIGCOMM Conf. SIGCOMM*, Aug. 2013, pp. 39–50.

[22] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith, "Inter-technology backscatter: Towards internet connectivity for implanted devices," in *Proc. ACM SIGCOMM Conf.*, Aug. 2016, pp. 356–369.

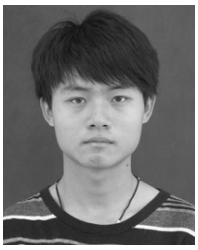
[23] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-Fi backscatter: Internet connectivity for RF-powered devices," in *Proc. ACM Conf. SIGCOMM*, Aug. 2014, pp. 607–618.

[24] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti, "BackFi: High throughput WiFi backscatter," in *Proc. ACM Conf. Special Interest Group Data Commun.*, Aug. 2015, pp. 283–296.

[25] P. Zhang, M. Rostami, P. Hu, and D. Ganesan, "Enabling practical backscatter communication for on-body sensors," in *Proc. ACM SIGCOMM Conf.*, Aug. 2016, pp. 370–383.
[26] 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.
[27] Y. Peng et al., "PLoRa: A passive long-range data network from ambient LoRa transmissions," in *Proc. ACM SIGCOMM*, 2018, pp. 147–160.
[28] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "LoRa backscatter: Enabling the vision of ubiquitous connectivity," in *Proc. ACM IMWUT*, 2017, pp. 1–24.
[29] P. Zhang, D. Bharadia, K. Joshi, and S. Katti, "HitchHike: Practical backscatter using commodity WiFi," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2016, pp. 259–271.
[30] P. Zhang, C. Josephson, D. Bharadia, and S. Katti, "FreeRider: Backscatter communication using commodity radios," in *Proc. 13th Int. Conf. Emerg. Netw. EXperiments Technol.*, Nov. 2017, pp. 389–401.
[31] J. Zhao, W. Gong, and J. Liu, "X-tandem: Towards multi-hop backscatter communication with commodity WiFi," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2018, pp. 497–511.
[32] G. Vougioukas and A. Bletsas, "Switching frequency techniques for universal ambient backscatter networking," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 464–477, Feb. 2019.
[33] *IGLOO Nano*. Accessed: Oct. 17, 2022. [Online]. Available: https://www.microchip.com/en-us/products/fpgas-and-plds/fpgas/igloo-fpgas#
[34] A. Lozano-Nieto, *RFID Design Fundamentals and Applications*. Boca Raton, FL, USA: CRC Press, 2017.
[35] A. P. Sample, D. J. Yeager, P. S. Powledge, A. V. Mamishev, and J. R. Smith, "Design of an RFID-based battery-free programmable sensing platform," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 11, pp. 2608–2615, Nov. 2008.
[36] X. Zhang and K. G. Shin, "E-MiLi: Energy-minimizing idle listening in wireless networks," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 205–216.
[37] F. Lu, P. Ling, G. M. Voelker, and A. C. Snoeren, "Enfold: Downclocking OFDM in WiFi," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2014, pp. 129–140.
[38] F. Lu, G. M. Voelker, and A. C. Snoeren, "SloMo: Downclocking WiFi communication," in *Proc. USENIX NSDI*, 2013, pp. 255–268.
[39] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[40] S. Tarkoma, *Overlay Networks: Toward Information Networking*. New York, NY, USA: Auerbach, 2010.
[41] *ZigBee*. Accessed: Oct. 17, 2022. [Online]. Available: https://csa-iot.org/all-solutions/zigbee/
[42] S. W. Smith et al., *The Scientist and Engineer's Guide to Digital Signal Processing*, 1st ed. San Diego, CA, USA: California Technical Publishing, Jan. 1997.
[43] *Bluetooth*. Accessed: Oct. 17, 2022. [Online]. Available: https://www.bluetooth.com/specifications/
[44] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.
[45] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[46] S. Lee, A. P. Chandrakasan, and H. Lee, "A 12 b 5-to-50 MS/s 0.5-to-1 V voltage scalable zero-crossing based pipelined ADC," *IEEE J. Solid-State Circuits*, vol. 47, no. 7, p. 1603—1614, May 2012.
[47] K. Wan et al., "A 0.6 V 12 b 10 MS/s low-noise asynchronous SAR-assisted time-interleaved SAR (SATI-SAR) ADC," *IEEE J. Solid-State Circuits*, vol. 51, no. 8, pp. 1826–1839, Jun. 2016.
[48] J. Kimionis, A. Bletsas, and J. N. Sahalos, "Increased range bistatic scatter radio," *IEEE Trans. Commun.*, vol. 62, no. 3, pp. 1091–1104, Mar. 2014.
[49] P. N. Alevizos, N. Fasarakis-Hilliard, K. Tountas, N. Agadakos, N. Kargas, and A. Bletsas, "Channel coding for increased range bistatic backscatter radio: Experimental results," in *Proc. IEEE RFID Technol. Appl. Conf. (RFID-TA)*, Sep. 2014, pp. 38–43.
[50] P. N. Alevizos, A. Bletsas, and G. N. Karystinos, "Noncoherent short packet detection and decoding for scatter radio sensor networking," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 2128–2140, May 2017.
[51] P. N. Alevizos, K. Tountas, and A. Bletsas, "Multistatic scatter radio sensor networks for extended coverage," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4522–4535, Jul. 2018.
[52] G. Vannucci, A. Bletsas, and D. Leigh, "A software-defined radio system for backscatter sensor networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2170–2179, Jun. 2008.



Longzhi Yuan (Student Member, IEEE) received the B.S. degree from the School of Information Science and Technology, University of Science and Technology of China, Anhui, China, in 2017, where he is currently pursuing the Ph.D. degree with the School of Data Science, under the supervision of Dr. Wei Gong. His research interests include wireless networks and the IoT.



Qiwei Wang (Student Member, IEEE) received the B.S. degree in computer science and technology from the University of Electronic Science and Technology of China, Chengdu, China. He is currently pursuing the M.S. degree with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include networking, transport protocols, and the Internet of Things.

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 the 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.