Multiprotocol Backscatter for Personal IoT Sensors

Wei Gong[*]
University of Science and Technology of China
weigong@ustc.edu.cn

Longzhi Yuan
University of Science and Technology of China
longzhi@mail.ustc.edu.cn

Qiwei Wang
University of Science and Technology of China
qiweiw@mail.ustc.edu.cn

Jia Zhao
Simon Fraser University, Canada
zhaojiaz@sfu.ca

ABSTRACT

We present multiscatter, a novel 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. 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, and the maximal backscatter ranges are 28 m, 22 m, and 20 m for WiFi, Bluetooth, and ZigBee, respectively. We also demonstrate that it can leverage excitation diversity to provide uninterrupted communication and greater throughput gains, whereas the single-protocol tag being idle when target carriers are not available.

CCS CONCEPTS

• **Networks → Network design principles**;

KEYWORDS

backscatter, multiprotocol, WiFi, ZigBee, Bluetooth

ACM Reference Format:
Wei Gong, Longzhi Yuan, Qiwei Wang, and Jia Zhao. 2020. Multiprotocol Backscatter for Personal IoT Sensors. In *The 16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '20), December 1–4, 2020, Barcelona, Spain.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3386367.3432883

[*]Corresponding author: Wei Gong

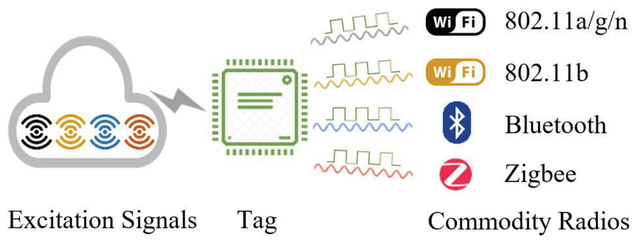


Figure 1: Multiscatter conceptual design. The multiscatter tag is able to identify different excitation signals and uses overlay modulation to convey tag data with productive data. Only a single commodity personal radio is adequate to decode both data.

1 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 [7–9, 11–16, 18, 27, 28, 31, 36, 37, 40, 46, 48]. 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 [6, 17, 19, 20, 23, 28, 29, 33, 39, 42–44, 47].

Along the line, 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 sensors.

Table 1: Comparison of backscatter systems.

	Excitation diversity	Productive carrier	Single commodity receiver
WiFi backscatter [19]		■	■
FS backscatter [44]		■	■
Interscatter [17]			■
Passive WiFi [20]			■
LoRa backscatter [33]			■
Hitchhike [42]		■	
FreeRider [43]		■	
X-Tandem [47]		■	
PLoRa [29]		■	
Multiscatter	■	■	■

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 [43], provides a holistic way to modulate multiple excitation signals, but does not support excitation diversity to distinguish different protocols simultaneously. Second, while packet-level backscatter systems, e.g., WiFi backscatter [19] and FS backscatter [44], 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 [17] and LoRa backscatter [33], only a single commodity receiver is needed, but the carrier has to be single tones generated by a Bluetooth device. In contrast, Hitchhike [42], FreeRider [43], and X-Tandem [47] can take any productive signals as carriers yet requires two synchronized receivers to decode the tag data. PLoRa [29], though supports productive carriers, cannot work with commodity receivers. A detailed comparison of state-of-the-art backscatter systems is shown in Table 1. 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 an ultra-low-power way?

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 baseband 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 §2.2 and §2.3.

(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 §2.4.

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. The maximal backscatter ranges are 28 m, 22 m, and 20m for WiFi, Bluetooth, 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 low-power 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 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.

2 MULTISCATTER DESIGN

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

2.1 Overview

As shown in Figure 2, the tag harvests RF power from abundant excitation signals, namely WiFi, Bluetooth, and ZigBee in the 2.4 GHz ISM band. Afterward, it uses a high-bandwidth rectifier to acquire baseband signals and correlates sampled bits with pre-stored templates. 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 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 [2]. Furthermore, decoding tag data should be easily done by a single commodity radio, e.g., Bluetooth on smartphones, for personal IoT applications.

2.2 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?

2.2.1 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. Traditionally, the RFID tag

[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.

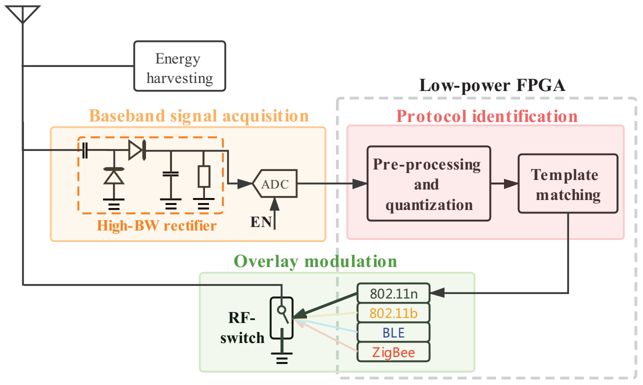


Figure 2: Multiscatter overview. Besides energy-harvesting, it first obtains baseband signal through the high-bandwidth rectifier, then identifies the incoming carrier type, and finally modulates tag data onto the carrier.

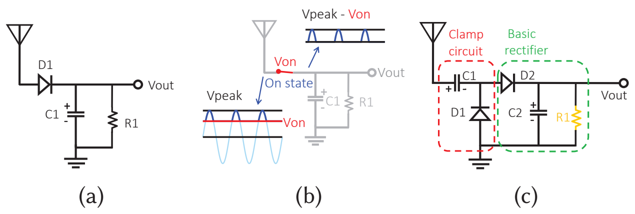
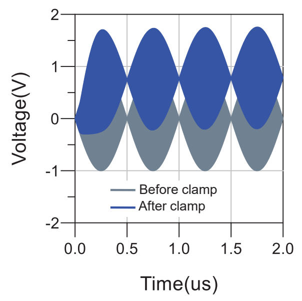


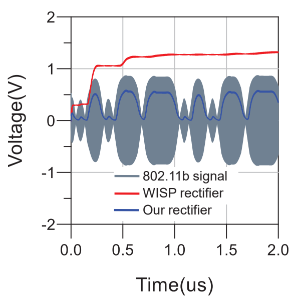
Figure 3: (a) Basic rectifier; (b) Voltage difference; (c) Our rectifier.

employs a basic rectifier to do so. As shown in Figure 3a, when the incoming signal's amplitude increases, the capacitor voltage is increased via the rectifying diode. When the input amplitude falls, the capacitor voltage is reduced as it is discharged to the resistor R_1. It is also called the "envelope detector" as the output is the *absolute value of the baseband signal.* This form is a cheap, simple, and effective solution to demodulate ASK (Amplitude Shift Keying) signals for RFID protocols. Nevertheless, lots of information gets lost for frequency- and phase-modulated signals.

To acquire quality basebands from high-bandwidth signals, e.g., WiFi, there are two key issues of using this rectifier. The first one is that the output voltage of the basic rectifier is not the peak voltage of the input but the difference between the peak voltage (V_{peak}) and the turn-on voltage of the diode (V_{on}), as shown in Figure 3b. So if the incoming peak voltage is less than the diode turn-on voltage, the diode will never turn on, which means no signal can come through. Meanwhile, the peak voltage squeezed from the tag's antenna is limited because high-frequency signals decay quickly in the air. To address this, we employ a clamp circuit, as shown in Figure 3c. When the input amplitude falls, due to the low turn-on voltage of diode D_1, V_{D1}, the voltage at the output of the clamp diode would not be lower than -V_{D1}; otherwise the GND can charge C_1 quickly through D_1, keeping the voltage no lower than -V_{D1}. As shown in Figure 4a, with an input signal at 2.4 GHz, the clamp circuit

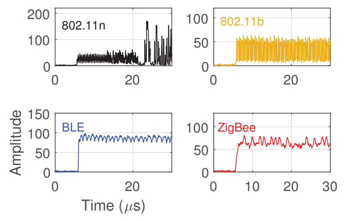


(a) Clamp voltage booster

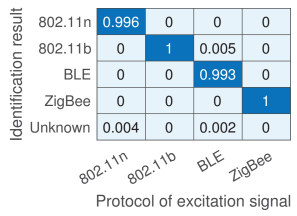


(b) Rectified signals

Figure 4: With the help of optimized clamp and RC circuits, our rectifier is able to obtain high-bandwidth baseband signals from 2.4 GHz carriers.



(a) Acquired baseband signals



(b) Identification accuracy

Figure 5: 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 $L_t = 120, L_p = 40$.

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 [24]. The second issue is the response rate of the rectifier. From Figure 3a, we know the quality of baseband signals from the rectifier is highly related to the discharging speed of the capacitor, which is τ, the time constant of an RC circuit. 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 $\frac{1}{f_c} \ll \tau \ll \frac{1}{f_b}$ [24]. In our case, $f_c = 2.4$ GHz and $f_b = 20$ MHz as bandwidths of Bluetooth and ZigBee are even lower. We compare our rectifier against WISP implementation [3]; results are shown in Figure 4b. The baseband signal of the WISP rectifier experienced large distortions for 802.11b input because it is designed for low-bandwidth signals from RFID readers, which is typically between 40 to 160 kbps. In contrast, our rectifier fits the high-bandwidth input signal much better thanks to the properly tuned clamp and RC circuits.

To examine our rectifier's performance, we set the transmission power of 802.11n signals at 30 dBm [3], the output voltage threshold of our rectifier at 0.15 V, and tag sensitivity at -13 dBm (typically -9$\sim$-12 dBm for RFID tags); the achieved maximal downlink range is 0.9 m, which is less than the typical RFID reading range, $\approx$10 m. There are three main contributing factors for such a reduced downlink range. First, our rectifier has lower SNRs because it trades the output voltage (SNR) of the rectifier for fine-grained (high-frequency) baseband signals mainly due to the tuned resistor R_1. From Figure 4b, the output voltage of our rectifier is less than half of WISP, which translates to a 6 dB loss. Second, our target signals at 2.4 GHz have shorter wavelengths ($\approx$0.12 m) than RFID ($\approx$0.33 m), which brings less than 15% of the received energy for an RFID tag along the same path. It is an additional 8 dB drop approximately. Third, we use a typical personal WiFi device that has a 3 dBi omni-directional antenna whereas an RFID reader is usually with directional patched antennas of 9 dBic. Yet, for personal on-body sensors, 0.9 m downlink range is adequate to reuse excitation signals from smartwatches, cellphones, and laptops.

2.2.2 *Template Matching.* Despite the seemingly limited range of the downlink, our rectifier does bring us high-bandwidth baseband signals for identification. As shown in Figure 5a, 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 template size is L_t. It should have two parts: a preprocessing window of size L_p and a matching window of size L_m. The preprocessing window is for DC removal and normalization; the matching window is for correlation computation. 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 5b 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.

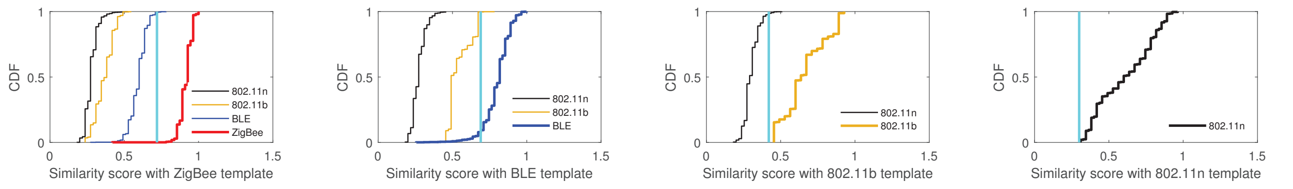
2.3 Low-Power Protocol Identification

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

2.3.1 *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

[3]30 dBm power level is achieved by using a power amplifier

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

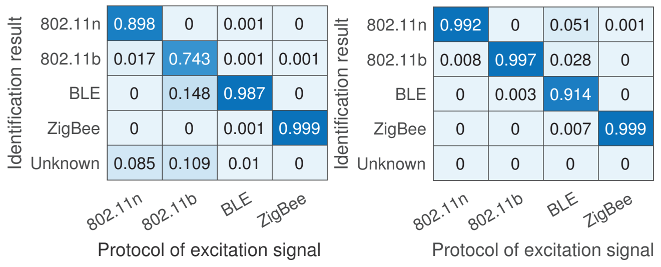


(a) Step 1: $C_{\text{Zigbee}} > Thrs_z$ (b) Step 2: $C_{\text{BLE}} > Thrs_b$ (c) Step 3: $C_{802.11\text{b}} > Thrs_{11b}$ (d) Step 4: $C_{802.11\text{n}} > Thrs_{11n}$

Figure 6: Our ordered matching, from ZigBee to BLE, to 802.11b, and to 802.11n.

Table 2: Comparison of different FPGA implementations for multiprotocol identification.

	Multipliers	Adders	D-Flip-Flops
802.11n[4]	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

Figure 7: Comparison of blind and ordered matching at a sampling rate of 10 Msps with quantization.

D-Flip-Flops consumed would be 133,364 as shown in Table 2. This is too many because an AGLN250 has only 6,144 D-Flip-Flops.

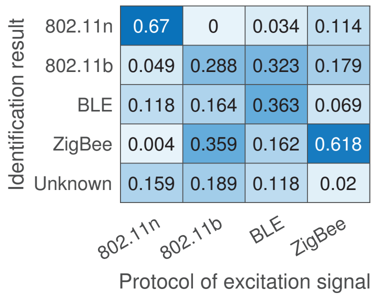
To address this, our solution is to do quantization and downsampling [25, 26, 45] at the same time. As quantization is a lossy process that reduces the precision of samples, we trade identification accuracy for meeting ultra-lower-power FPGA requirements. Specifically, we quantize each sample into 1 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 7a. Compared to Figure 5b, we observe that quantization and downsampling do degrade detection accuracy but not too much.

2.3.2 *Ordered Matching.* During the above lossy process, we have another interesting observation: four excitation signals demonstrate noticeably different resilience. For example, as shown in 6a, 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 6 and the corresponding accuracy results are demonstrated in Figure 7b. 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.

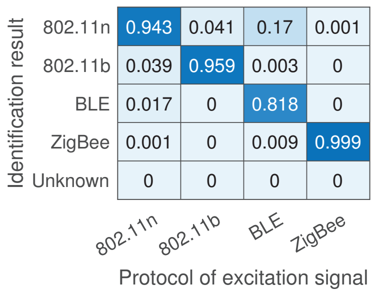
With the help of ordered matching, we attempt to keep reducing sampling rates and find that when the sampling rate is 2.5 Msps, it becomes tough to differentiate the four signals. The average identification accuracy is only 0.485 as shown in Figure 8a. Therefore, we intend to prolong the matching window, i.e., finding the maximal matching window size. We observe that only BLE and 802.11n are the limiting factors 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 8b. 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 8c.

A few points are worth noting:

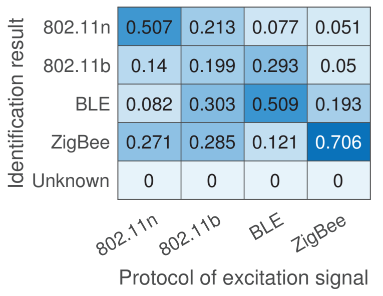
(1) We have the EN-signal for the ADC, which is controlled by the FPGA to do duty-cycling and thus avoid excessive



(a) 8μs-matching window at 2.5 Msps



(b) 40μs-matching window at 2.5 Msps



(c) 40μs-matching window at 1 Msps

Figure 8: 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.

power consumption. Further power saving can be made by introducing an additional wake-up module, like [30].

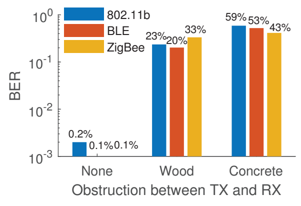
(2) Although our ultra-low-power FPGA supports a limited storage space of 36 kb for both code and data, the storage overhead of four templates with the extended length is 400 bits, which only costs 1.1% of the total storage space.

(3) 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.

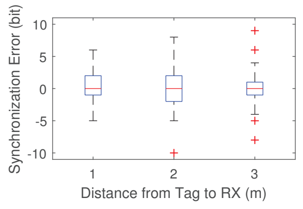
2.4 Overlay Modulation

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.

2.4.1 Motivation. Being able to handle productive-data carriers is an important feature for backscatter, as neither dedicated RFID readers nor single-tone generators [17] 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 [29, 43, 46] 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. We perform experiments for Hitchhike and FreeRider in three scenarios: the original channels with no obstruction, a wooden wall, and a concrete wall. As shown in Figure 9a, the BER of tag data through 802.11b carriers is increased from 0.2% for non-occlusion to 59% for a concrete wall blocking the original channel. 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. Figure 9b shows that Hitchhike has large modulation offsets, as far as 8 bits (symbols), across different ranges. Such large offsets happen because currently there is no scheme on the tag



(a) BER vs occlusion



(b) Tag modulation offset

Figure 9: Hitchhike and FreeRider suffer from two problems: 1) severe BER degradation when the original channel is occluded; 2) significant modulation offsets.

that can accurately synchronize (symbol-level) itself with the WiFi carrier. 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.

2.4.2 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 [35] and the overlay network that is built on top of another network [34]. The detailed workflow is as follows. As shown in Figure 10, 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 time,

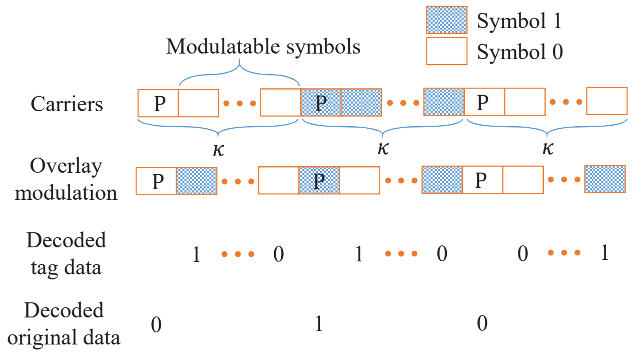


Figure 10: Overlay modulation where the carrier is composed of a couple of modulatable sequences. Each sequence has a reference symbol carrying productive data and modulatable symbols for tag data.

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 [42] and FreeRider [43], we will discuss how to do tag-data modulation and demodulation with WiFi [5], ZigBee, and Bluetooth as follows.

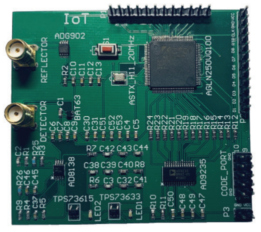
802.11b. For 802.11b excitation signals, reference symbols support DSSS-BPSK, DSSS-DQPSK, and CCK modulation (same as original 802.11b modulation). Despite various modulation schemes for reference symbols, we observe that BPSK-based tag-data modulation is compatible with all of them. Specifically, if the excitation signal is identified as 802.11b, we first frequency shift it to another channel [6] and thus avoid creating interference in the original channel [17, 44] [7]. 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 Miller code in RFID that uses long modulation length to combat low SNRs [5], 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

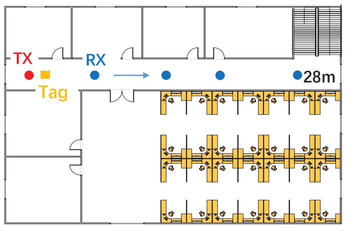
[5]Currently, we mainly focus on two types of WiFi: (1)DSSS and CCK modulation: 802.11b, and (2)the OFDM modulation that covers 802.11a/g/n/ac/ax.

[6]It is possible that we shift to a busy channel. Addressing this problem requires channel sensing, which is not supported by most backscatter tags.

[7]We perform center-frequency alignment by a brute-force search.



(a) Multiscatter tag prototype.



(b) Experiment deployment.

Figure 11: Our tag prototype, around \$60/each, and the experimental area is 30m*50m.

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 [8].

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 [32], i.e., BPSK-based tag-data modulation stands. The difference is the unit of tag-data modulation becomes OFDM symbol, 4μs for each. Another thing is that as the scrambler and BCC encoder are not completely compatible with codeword translation [43], which may leads 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 [46].

ZigBee. Reference-symbol modulation for ZigBee adopts offset quadrature phase-shift keying (OQPSK) [4]. 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 maybe not as expected, 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 [1], 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

[8]Similar to prior works, e.g., HitchHike and FreeRider, the reason of choosing majority voting and repetition coding is that they are simple and efficient. Our future work includes the investigation of more sophisticated coding schemes, e.g., Forward Error Correction (FEC).

Table 3: Power consumption of our COTS prototype

Logical part	Devices	Power(mW)
Pkt det.	Pkt det.(FPGA)	2.5
	ADC (20 Msps)	260
Modulation	FPGA (Modulation)	1.0
	RF-switch	0.1
Clock	Oscillator (20 MHz)	15.9
Total		279.5

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.

2.4.3 Summary. While the way of modulating symbols is inspired by codeword translation [42, 43], our overlay modulation is built on top of it and beyond. The major differences are as follows:

1) 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.

2) The spectral efficiency is largely improved as the required decoding spectrum is the same as the original channel, whereas prior work [29, 42, 43, 46, 47] demands twice of that.

3) 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.

3 IMPLEMENTATION

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

Off-the-Shelf Prototype. Our tag prototype consists of two main parts: an RF front-end and an FPGA for baseband processing. As shown in Figure 11a, there are two antennas in the front-end. One is connected to the envelope detector circuit and ADC, for multiprotocol identification. Our envelope detector circuit is simple, which has only diodes and capacitors. It removes high-frequency carrier and generates envelopes for further FPGA processing. An AD9235 ADC is used to sample baseband signals and sampled data is fed into the FPGA. The other backscatter antenna is connected to an ADG902 RF switch, and baseband processing is implemented using an Igloo nano AGLN250 FPGA. All the baseband processing, including DC removal and normalization, multiprotocol identification, phase and frequency modulation are realized in an AGLN250, which is ultra-low-power in nature and has only 6,144 D-flip-flops.

Power consumption. Although our prototype is designed mainly for function verification, we perform rigorous power analysis that contains the breakdown of peak power consumption (20 Msps) for all the parts, as shown in Table 3. There are three modules: packet detection, modulation, and clock, which consume 262.5 mW, 1.1 mW, and 15.9 mW, respectively. The total peak power consumption is 279.5 mW. While the power consumption of our current PCB prototype is mainly constrained by COTS electronic components, like ADC, there are two ways to improve.

First, advanced IC design promises to deliver much lower power consumption. In particular, to replace power hungry component AD9235, for future IC design, we can employ recent works on advanced ADCs that consumes only tens of μW at the sampling rate of tens of Msps [10, 21, 22, 38]. For example, [38] provides an implementation that consumes only 83 μW at a sampling rate of 10 Msps. In contrast, we only require a rate of 2.5 Msps. Furthermore, building IC-based baseband will bring additional power savings. We simulate all the functions of our baseband processing using Libera, a simulation software provided by the AGLN250 manufacturer. The results show that the baseband power consumption is 1.89 mW. Note that AGLN250 is manufactured using an old 130-nm CMOS process, so an even lower-power (at the scale of hundreds of μW) design using most advanced ASIC processes (5-7 nm) is quite promising.

Second, we can include more energy harvesters when RF-power is not enough to drive the tag. We have tested an MP3-37 solar panel and integrated it into our system. The harvested energy is managed by the TI power management chip BQ25570 and stored

Table 4: Average tag-data exchange times of a single packet under different lighting conditions where excitations rates are 2000 pkts/s for 802.1n and 802.11b, 70 pkts/s for BLE, and 20 pkts/s for ZigBee.

	Total time	Exchange packets	Average exchange time	
			Indoor	Outdoor
802.11n	Indoor:	360	0.60s	2.2ms
802.11b	217.2s	360	0.60s	2.2ms
BLE	Outdoor:	12.6	17.2s	61.9ms
ZigBee	0.78s	3.6	60.1s	21.7ms

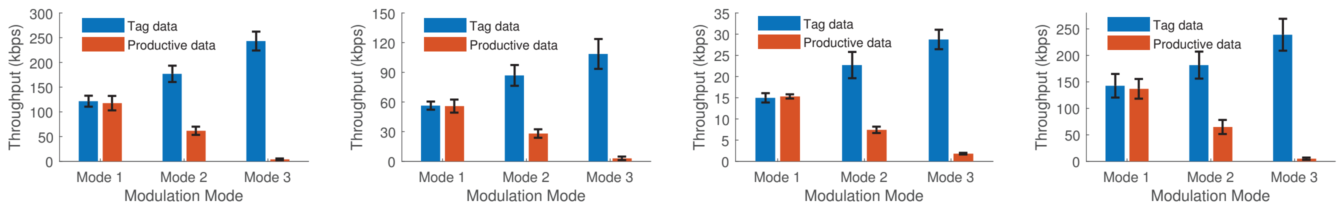
Table 5: Required hardware resources and power consumption of protocol-identification algorithms

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

Table 6: 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	$4n$ [9]
802.11n, $\gamma = 2$	4	8	$2n$
BLE, $\gamma = 4$	8	16	$4n$
ZigBee, $\gamma = 2$	4	8	$2n$

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



(a) WiFi 802.11b (b) WiFi 802.11n (c) ZigBee (d) Bluetooth

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

in a storage capacitor of 0.01F. When the voltage on the storage capacitor exceeds 4.1V, the power is ready to use for computation and communication parts. Then the voltage drops gradually and when it reaches 2.6V, the BQ25570 chip shuts down the power. Hence, the energy supplied by the storage capacitor for a discharging round is $\frac{1}{2}\times 0.01F\times((4.1V)^2)-(2.6V)^2) = 50mJ$. This energy can support the system to work for $50/279.5 = 0.18s$. Suppose that 802.11n, 802.11b, BLE, and ZigBee excitations signals are transmitted at 2000 pkts, 2000 pkts, 70 pkts, 20 pkts, respectively, then 50 mJ can support a multiscatter tag to backscatter 360 pkts for 802.11n, 360 pkts for 802.11b, 12.6 pkts for BLE, and 3.6 pkts for ZigBee. At the same time, to harvest 50 mJ, it takes 216.2 s and 0.78 s when the light strengths are of 500 Lux for indoor and 1.04×10^5 Lux for outdoor scenarios. Therefore, the average times of a single tag-data exchange for indoor cases with 802.11n, 802.11b, BLE, and ZigBee are 0.6 s, 0.6 s, 17.2 s, and 60.1 s, respectively, as shown 4. The counterparts for outdoor cases are 2.2 ms, 2.2 ms, 61.9 ms, and 21.6 ms.

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 an XILINX Artix-7 FPGA because variant implementations are too complex to deploy on an AGLN250. Results are shown in Table 5 [10]. 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× lower power than the naive implementation.

Experimental Setup. Figure 11b shows the floor plan. In the LoS scenarios, all devices are placed in the hallway; we deployed a multiscatter tag 0.8 m away from the commodity radios (802.11b/n WiFi, ZigBee, or Bluetooth), then we move the receiver away from the tag and measure the received signal strength indicator (RSSI), bit error rate (BER) and throughput of the backscattered signal. In the NLoS scenarios, we place the transmitter and the multiscatter tag in the office, and the receiver is still placed in the hallway, The rest of experimental settings is the same as LoS cases'.

For WiFi, we use Qualcomm Atheros AR938X NICs as both productive carrier generators and receivers, 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. 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. 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.

4 EVALUATION

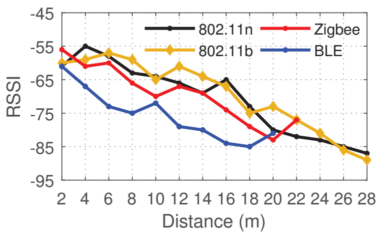
4.1 End-to-End Performance

In this study, we mainly answer the following questions: what kind of tradeoffs of productive and tag data can be made? what are the maximal backscatter ranges in LoS and NLoS cases? and what is the negative impact of the requirement of original packets in prior work?

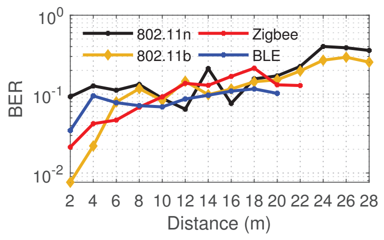
4.1.1 Tradeoffs between Productive and Tag Data. According to the design of overlay modulation, γ determines how long a reference symbol is and κ defines the number of modulatable symbols. Since the reference symbol carries productive data and modulatable symbols carry tag data, we can adjust ratios of the two to make tradeoffs. In particular, we define three modes as shown in Table 6. 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. γ values of four protocols are empirically chosen to achieve the best throughputs while maintaining BERs less than 10^{-1} for short distances.

In this experiment, we keep the transmitter and receiver stationary and move the tag at multiple locations to explore spatial diversity. We report the average value of 100 independent locations for all the four signals. The results 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 3x than that of reference symbols. Same observations can also be made for the

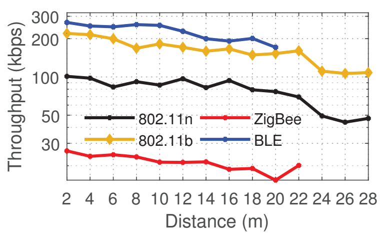
[10]LUT (Look-Up-Table) is the basic configurable logic element in an FPGA. Usually, the number of LUTs used is proportional to the IC-simulation power consumption.



(a) RSSI

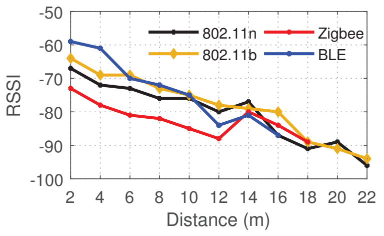


(b) BER

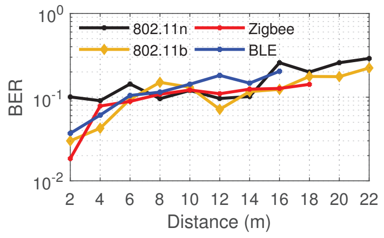


(c) Throughput

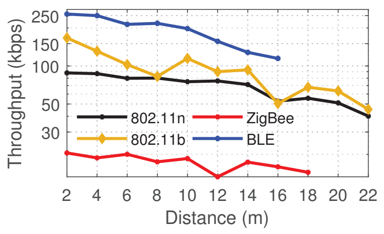
Figure 13: Backscatter RSSI, BER, and throughput across distances in LoS deployment.



(a) RSSI



(b) BER



(c) Throughput

Figure 14: Backscatter RSSI, BER, and throughput across distances in NLoS deployment.

rest protocols. In mode 3, as expected, we barely see any throughput for productive data. Because each packet can only carry 1-bit productive data. In contrast, tag data throughput is maximized compared to mode 1 and 2. To conclude, this experiment validates our overlay modulation design, which for the first time provides much flexibility to achieve tradeoffs between productive data and tag data. According to different application requirements, the tradeoff 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.

4.1.2 *Effective Backscatter Ranges.*

Next, we intend to examine the maximal backscatter ranges for excitation signals.

LoS scenarios: We first measure the RSSI, BER and throughput of multiscatter in the line-of-sight scenario. Figure 13a shows that the maximum backscatter communication range of WiFi(11b/n), ZigBee and Bluetooth are 28 m, 22 m, and 20 m, respectively. From Figure 13b, 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 13c 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. As the distance increases, the backscattered signals' packet rate drops gradually. For example, the receiver hardly receives backscattered packets when the distance is longer than 20 m for BLE excitation signals.

NLoS scenarios: We also evaluate multiscatter in non-line-of-sight deployment. As shown in Figure 14a, the maximum NLoS backscatter communication range of WiFi(11b/n), ZigBee and Bluetooth are 22 m, 18 m, and 16 m, respectively. Specifically, WiFi and BLE receivers can achieve a signal strength of more than -75 dBm within

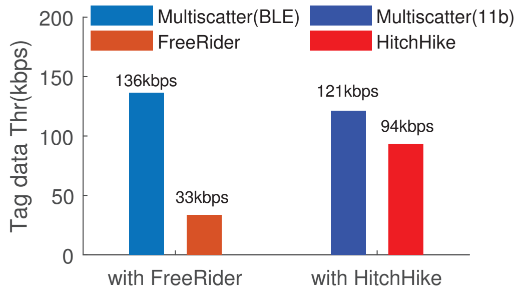
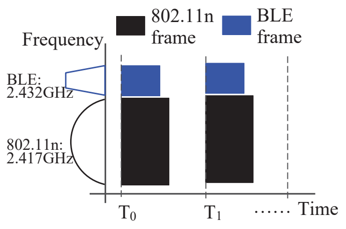


Figure 15: Comparison of tag-data throughputs when there is an occlusion in the original channel.

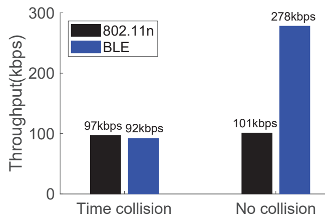
10 m, while ZigBee signal strength reduces quickly to less than -80 dBm if the distance exceeds 4 m. As expected, multiscatter's performance degrades across all protocols with NLoS scenarios but still manifests its resilient ability to work with occlusions.

4.1.3 *Negative Impact of Requiring Original Packets.*

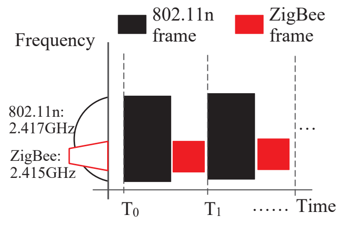
Previous backscatter systems [29, 42, 43, 47] require two receivers to decode tag data, where one receiver works on the original packet and another captures the backscattered packet. These systems base their decoding function on a precondition: the excitation signal is always well received and 100% correctly decoded. However, it is not always the case. We conduct experiments to examine this impact and find that even when the original channel is occluded by a thin drywall, the original data reception becomes highly unstable. The detailed results are depicted in Figure 15. For tag-data



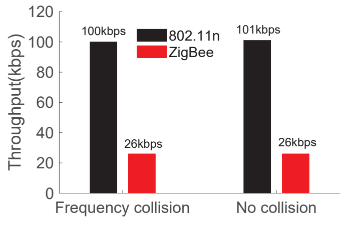
(a) Excitations collide in time.



(b) Throughput comparison of with and w/o time collisions.



(c) Excitations collide in frequency.



(d) Throughput comparison of with and w/o frequency collisions.

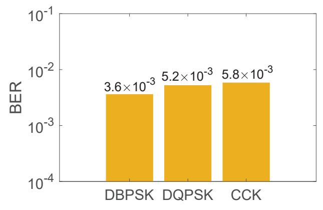
Figure 16: Throughput performance when diverse excitation carriers collide in the time domain and frequency domains.

throughput, multiscatter achieves 136 kbps and 121 kbps for BLE and 802.11b excitations thanks to our single-receiver design while the throughput of FreeRider is 33 kbps and that of Hitchhike is 94 kbps. Hence, in spite of occlusions, multiscatter still achieves better tag-data throughput than both FreeRider and Hitchhike. The main contributing factor is that low-quality data from the original channel significantly impacts the tag data decoding. What's worse, if original packets are completely lost, backscattered packets cannot be decoded correctly at all. In contrast, multiscatter does not require original packets for decoding and thus avoids this problem. Both productive and tag data can be decoded by multiscatter using only a single commodity receiver.

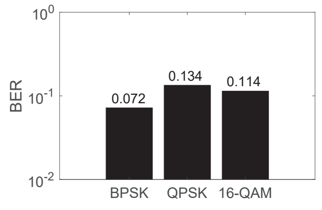
4.1.4 Impact of Collided Excitations. Next, we examine how multiscatter performs when different excitations collided in the time and frequency domains. For experimental setup, we take mode 1 where the ratio of modulatable symbols to reference symbols is 1:1 and report aggregated throughput. As shown in Figure 16a, we employ two excitations collided in time. The 802.11n excitation is transmitted at 2.417 GHz with 2000 pkts/s. Each packet is 300-byte long. The other excitation is BLE signal at 2.432 GHz with 34 pkts/s [11]. Two excitations collided in time. And since multiscatter does not employ filters which are heavily used in active commercial radios, both excitations would collide on the tag. The results are shown in Figure 16b. We observe that the throughput for 802.11n does not change too much while the throughput for BLE experiences a evident drop from 278 kbps to 92 kbps. This is because our excitation-collision setup in Figure 16a is close to real scenario where WiFi packets are more intense and longer than BLE packets. Thus, when two carriers collides, both would take throughput losses. But such losses only makes only slight differences in the overall throughput for WiFi because WiFi excitation is sent at a rate of 2000 pkts/s, which is way higher than BLE's 34 pkts/s. Therefore, to protect BLE throughput in such scenarios, filters on the tag would be necessary, which we leave for future work.

Also, we investigate two excitations collided in frequency, as shown in Figure 16c. The 802.11n excitation stays the same while the other excitation is ZigBee signal at 2.415 GHz with 20 pkts/s.

[11]The current COTS BLE chips only support full control over advertising packets and its maximum transmission rate is 70 pkts/s due to advertising intervals and unknown link-layer delay [41] and the packet length is 37 bytes. The measured advertising rates on our campus (including classroom, lab, library, cafeteria) are mostly between 30 pkts and 40 pkts. So we set the BLE excitation rate at 34 pkts/s to simulate real-world scenarios in our controlled experiments.



(a) 802.11b ref-sym modulation.



(b) 802.11n ref-sym modulation.

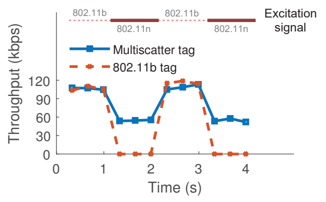
Figure 17: BERs with various reference-symbol modulation schemes.

Each ZigBee packet is 200-byte long. We depict results in Figure 16d. The observation is that both ZigBee and 802.11n throughputs are not much affected by collisions in frequency. This is primarily due to that both excitations are not overlapped in the time domain and our multiprotocol identification design can effectively distinguish such packets thanks to ordered template matching.

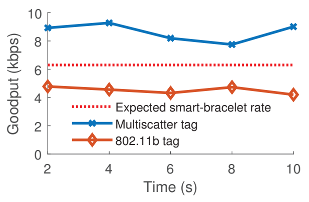
In a word, our current design is fully compatible with TDMA-like multiprotocol excitations and plan to handle FDMA-like multiprotocol excitations better in the future.

4.1.5 Impact of Reference-Symbol Modulation. Next, we are going to examine the impact of different reference-symbol modulation schemes on tag-data communication. Since both ZigBee and Bluetooth only support single modulation patterns, OQPSK for ZigBee and GFSK for Bluetooth, here we focus on 802.11b and 802.11n protocols. Specifically, we investigate BERs of tag data with reference symbols modulated with DSSS-BPSK, DSSS-DQPSK, and CCK (5.5Mbps) for 802.11b, and OFDM-BPSK, OFDM-QPSK, and OFDM-16QAM for 802.11n.

As shown in Figure 17a and 17b, BERs of tag data with different modulation schemes are quite stable and within a good range. In particular, for 802.11b, BERs are all below 0.6% with all different modulations. Those results show that our overlay modulation are robust across different reference-symbol modulation schemes. Further, employing such reference symbols causes no noticeable negative effects on channel estimation or signal acquisition. Because reference symbols are in the payload part and channel estimation is mainly done in the preamble part. If not, it is difficult to observe



(a) Uninterrupted backscatter using multiple excitations.



(b) Intelligent carrier pick.

Figure 18: Performance comparison of multiscatter tags with single-protocol tags.

stable and decent BER results for both 802.11b and 802.11n in Figure 17a and 17b.

4.2 Leveraging Excitation Diversity

4.2.1 Adaptation to Discontinuous Excitations. In practical applications, it is common that a specific signal we expect does not always exist, and different types of excitation signals may appear intermittently. First, we create periodical 802.11b and 802.11n productive carriers as shown in Figure 18a, each taking 50% duty cycle. Then we measure the throughputs for both a multiscatter and an 802.11n tag. As shown in Figure 18a, the multiscatter tag is busy transmitting data all the time, while the 802.11b tag is idle for 50% of the testing time. Such a huge difference comes from that the multiscatter tag is able to identify both 802.11n and 802.11b excitation signals and leverages them to do transmission accordingly, which clearly shows its ability to exploit excitation diversity.

4.2.2 Mixed Excitation Signals. For a backscatter system working only with one protocol, the data-rate is difficult to improve even if there are multiple available excitation signals. 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 18b, after evaluating the excitation rates of all signals, our 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.

5 RELATED WORK

For the last decade, turning backscatter into general-purpose communication for IoT networks has been a hot topic in the wireless network community. The seminal work, ambient backscatter [23], creatively proposes to use ambient TV signals as carriers to enable device-to-device backscatter communication for the first time. It opens the door for backscatter communication of reusing existing signals. FS backscatter [44] observes that frequency-shifting is the key to improve SNRs. BackFi [6] improves backscatter throughput to high data rates of 5-300 Mbps. Despite its high throughput, the required full-duplex radios for self-interference cancellation are hard to realize for off-the-shelf devices. As a result, the community looks into how to achieve symbol-level backscatter that can communicate with commodity radios.

Passive WiFi [20] 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 [33] and BLE-backscatter. FM backscatter [39] is proposed to creates backscatter transmissions decodable on any FM receiver. Nevertheless, the raw FM receiver becomes obsolete and most up-to-date smartphones do not support it many years ago. To overcome the limitation of infrastructure support, interscatter [17] comes in. It is the first work that enables high data rate backscatter using only commodity devices. 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 [42] is the first work that enables productive backscatter with commodity devices at the symbol level. The major contribution is codeword translation that enables symbol-level backscatter modulation by changing one codeword into another (valid) codeword. Enabling such productive backscatter significantly widens backscatter sources and makes ubiquitous backscatter vision closer. FreeRider [43], X-Tandem [47], PLoRa [29] 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 carriers 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.

6 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.

ACKNOWLEDGEMENT

We thank the anonymous shepherd and reviewers for their helpful comments. This work was supported by NSFC Grant No. 61932017 and 61971390, and the Fundamental Research Funds for the Central Universities No. WK2150110013.

REFERENCES

[1] [n. d.]. Bluetooth. https://www.bluetooth.com/specifications/bluetooth-core-specification/. ([n. d.]).

[2] [n. d.]. Igloo nano. https://www.microsemi.com/product-directory/fpgas/1689-igloo#igloo-nano. ([n. d.]).

[3] [n. d.]. WISP 5.0. https://github.com/wisp/wisp5. ([n. d.]).

[4] [n. d.]. ZigBee. https://zigbee.org/zigbee-for-developers/zigbee-3-0/. ([n. d.]).

[5] 2017. EPC C1G2 Standard. http://www.gs1.org/epcrfid/epc-rfid-uhf-air-interface-protocol/2-0-1. (2017).

[6] D. Bharadia, K. Joshi, M. Kotaru, and S. Katti. 2015. Backfi: High throughput wifi backscatter. In *Proc. of ACM SIGCOMM*.

[7] S. Chen, W. Gong, J. Zhao, and J. Liu. 2020. High-Throughput and Robust Rate Adaptation for Backscatter Networks. *IEEE/ACM Transactions on Networking* 28, 5 (2020), 2120–2131.

[8] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan. 2017. Empowering low-power wide area networks in urban settings. In *Proc. of ACM SIGCOMM*. ACM, 309–321.

[9] J. F. Ensworth and M. S. Reynolds. 2017. BLE-backscatter: Ultralow-power IoT nodes compatible with bluetooth 4.0 low energy (BLE) smartphones and tablets. *IEEE Transactions on Microwave Theory and Techniques* 65, 9 (2017), 3360–3368.

[10] M. Eslami, M. Taherzadeh-Sani, and F. Nabki. 2015. A 1-V 690μsW 8-bit 200 MS/s flash-SAR ADC with pipelined operation of flash and SAR ADCs in 0.13μsm CMOS. In *IEEE International Symposium on Circuits and Systems*.

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

[12] W. Gong, S. Chen, J. Liu, and Z. Wang. 2018. MobiRate:Mobility-Aware Rate Adaptation Using PHY Information for Backscatter Networks. In *Proc. of IEEE INFOCOM*.

[13] W. Gong, H. Liu, J. Liu, X. Fan, K. Liu, Q. Ma, and X. Ji. 2018. Channel-aware rate adaptation for backscatter networks. *IEEE/ACM Transactions on Networking* 26, 2 (2018), 751–764.

[14] W. Gong, J. Liu, K. Liu, and Y. Liu. 2016. Toward more rigorous and practical cardinality estimation for large-scale RFID systems. *IEEE/ACM Transactions on Networking* 25, 3 (2016), 1347–1358.

[15] M. Hessar, A. Najafi, and S. Gollakota. 2019. NetScatter: Enabling Large-Scale Backscatter Networks.. In *Proc. of USENIX NSDI*.

[16] P. Hu, P. Zhang, M. Rostami, and D. Ganesan. 2016. Braidio: An integrated active-passive radio for mobile devices with asymmetric energy budgets. In *Proc. of ACM SIGCOMM*.

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

[18] J. Jang and F. Adib. 2019. Underwater Backscatter Networking. In *Proc. of ACM SIGCOMM*.

[19] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall. 2014. Wi-Fi backscatter: Internet connectivity for RF-powered devices. In *Proc. of ACM SIGCOMM*.

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

[21] S. Lee, Anantha P. Chandrakasan, and H Lee. 2012. A 12 b 5-to-50 MS/s 0.5-to-1 V Voltage Scalable Zero-Crossing Based Pipelined ADC. *IEEE Journal of Solid-State Circuits* 47, 7 (2012), 1603—1614.

[22] C. Liu, M. Huang, and Y. Tu. 2016. A 12 bit 100 MS/s SAR-Assisted Digital-Slope ADC. *IEEE Journal of Solid-State Circuits* PP, 99 (2016), 1–10.

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

[24] Albert Lozano-Nieto. 2017. *RFID design fundamentals and applications*. CRC press.

[25] F. Lu, P. Ling, G. M. Voelker, and A. C. Snoeren. 2014. Enfold: downclocking ofdm in wifi. In *Proc. of ACM MOBICOM*.

[26] F. Lu, G. M. Voelker, and A. C. Snoeren. 2013. Slomo: Downclocking wifi communication. In *Proc. of USENIX NSDI*.

[27] Y. Ma, Z. Luo, C. Steiger, G. Traverso, and F. Adib. 2018. Enabling deep-tissue networking for miniature medical devices. In *Proc. of ACM SIGCOMM*.

[28] S. Naderiparizi, M. Hessar, V. Talla, S. Gollakota, and J. R. Smith. 2018. Towards battery-free HD video streaming. In *Proc. of USENIX NSDI*.

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

[30] N. E. Roberts, K. Craig, A. Shrivastava, S. N. Wooters, Y. Shakhsheer, B. H. Calhoun, and D. D. Wentzloff. 2016. 26.8 A 236nW 56.5dBm sensitivity bluetooth low-energy wakeup receiver with energy harvesting in 65nm CMOS. In *Proc. of IEEE ISSCC*.

[31] M. Rostami, J. Gummeson, A. Kiaghadi, and D. Ganesan. 2018. Polymorphic radios: A new design paradigm for ultra-low power communication. In *Proc. of ACM SIGCOMM*.

[32] Steven W Smith et al. 1997. The scientist and engineer's guide to digital signal processing. (1997).

[33] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota. 2017. Lora backscatter: Enabling the vision of ubiquitous connectivity. In *Proc. of ACM IMWUT*.

[34] S. Tarkoma. 2010. *Overlay Networks: Toward Information Networking*. Auerbach Publications.

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

[36] A. Varshney, O. Harms, C. Pérez-Penichet, C. Rohner, F. Hermans, and T. Voigt. 2017. Lorea: A backscatter architecture that achieves a long communication range. In *Proc. of ACM SENSYS*.

[37] D. Vasisht, G. Zhang, O. Abari, H. Lu, J. Flanz, and D. Katabi. 2018. In-body backscatter communication and localization. In *Proc. of ACM SIGCOMM*.

[38] K. Wan, H. Hong, Y. Roh, H. Kang, S. Hwang, D. Jo, D. Chang, M. Seo, and S. Ryu. 2016. A 0.6 V 12 b 10 MS/s Low-Noise Asynchronous SAR-Assisted Time-Interleaved SAR (SATI-SAR) ADC. *IEEE Journal of Solid-State Circuits* 51, 8 (2016), 1826–1839.

[39] A. Wang, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota. 2017. FM Backscatter: Enabling Connected Cities and Smart Fabrics. In *Proc. of USENIX NSDI*.

[40] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. 2012. Efficient and reliable low-power backscatter networks. In *Proc. of ACM SIGCOMM*.

[41] M. Zhang, J. Zhao, S. Chen, and W. Gong. 2020. Reliable Backscatter with Commodity BLE. In *Proc. of IEEE INFOCOM*.

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

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

[44] P. Zhang, M. Rostami, P. Hu, and D. Ganesan. 2016. Enabling practical backscatter communication for on-body sensors. In *Proc. of ACM SIGCOMM*.

[45] X. Zhang and K. G. Shin. 2011. E-MiLi: Energy-Minimizing Idle Listening in Wireless Networks. In *Proc. of ACM MOBICOM*.

[46] J. Zhao, W. Gong, and J. Liu. 2018. Spatial Stream Backscatter Using Commodity WiFi. In *Proc. of ACM MobiSys*.

[47] J. Zhao, W. Gong, and J. Liu. 2018. X-tandem: Towards multi-hop backscatter communication with commodity wifi. In *Proc. of ACM MOBICOM*.

[48] J. Zhao, W. Gong, and J. Liu. 2020. Towards Scalable Backscatter Sensor Mesh with Decodable Relay and Distributed Excitation. In *Proc. of ACM SENSYS*.