

PassiveBLE: Towards Fully Commodity-Compatible BLE Backscatter

Huixin Dong[†], Yijie Wu[†], Feiyu Li[†], Wei Kuang[†], Yuan He[#], Qian Zhang[△], Wei Wang^{†*}

[†]Huazhong University of Science and Technology,

[#]Tsinghua University, [△]Hong Kong University of Science and Technology

{huixin, yijiewu, feiyuli, kuangwei, weiwangw}@hust.edu.cn, heyuan@tsinghua.edu.cn, qianzh@cse.ust.hk

Abstract

Bluetooth Low Energy (BLE) backscatter is a promising candidate for battery-free Internet of Things (IoT) applications. We propose PassiveBLE, a backscatter system that can establish authentic and fully compatible BLE connections on data channels. The key enabling techniques include (i) a synchronization circuit that can wake up tags and activate backscatter communications with symbol-level accuracy to facilitate BLE data packet generation; (ii) a distributed coding scheme that offloads the major encoding and processing burdens from tags to the excitation source while achieving high throughput; (iii) a BLE connection scheduler to enable fully compatible BLE connection interactions, including connection establishment, maintenance and termination for multiple backscatter tags. We prototype PassiveBLE tags with off-the-shelf components and also conduct comprehensive experiments to evaluate the performance. Results demonstrate that PassiveBLE achieves a success rate of over 99.9% in establishing commodity BLE connections. PassiveBLE also achieves commodity-compatible BLE communication with a high goodput of up to 974 kbps in *LE 2M PHY* mode and 532 kbps in *LE 1M PHY* mode, which is about 63.3× higher than existing commodity-level BLE backscatter system in the same mode.

CCS Concepts

• **Networks** → **Network architectures**; • **Hardware** → **Networking hardware**.

*Corresponding author: Wei Wang.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. ACM MOBICOM '25, November 4–8, 2025, Hong Kong, China
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1129-9/2025/11

<https://doi.org/10.1145/3680207.3723465>

Keywords

Backscatter Technology; RF Computing; Internet of Things

ACM Reference Format:

Huixin Dong, Yijie Wu, Feiyu Li, Wei Kuang, Yuan He, Qian Zhan, Wei Wang. 2025. PassiveBLE: Towards Fully Commodity-Compatible BLE Backscatter. In *The 31st Annual International Conference on Mobile Computing and Networking (ACM MOBICOM '25)*, November 4–8, 2025, Hong Kong, China. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3680207.3723465>

1 Introduction

Both industry [3, 5, 34, 37] and academia [14, 22, 44–46] advocate the development of ultra-low-power Bluetooth Low Energy (BLE) backscatter to enable more power-efficient wireless communication in various applications, including logistics [2, 3, 5, 44], smart infrastructures [3, 37, 44], and wearable electronics [16, 37]. For example, the BLE chips typically consume 82.7% of the total power in an earbud [4]. Replacing it with a BLE backscatter would provide hundreds of kbps data rates while extending the battery life by approximately tenfold.

Existing BLE backscatter systems [14, 44–46] can only modulate data on advertising packets for one-shot communication. While promising, advertising-packet-based communication suffers from limited throughput, high delay, and security vulnerabilities. The inefficient modulation scheme in advertising packets limits the throughput of the commodity-compatible BLE backscatter to 8.4 kbps [44], while some BLE data collection applications, e.g., in-vehicle sensor networks [34], demand 125 kbps - 1 Mbps goodput. In addition, Tag activation in these systems relies on packet length detection, resulting in a delay of hundreds of milliseconds. However, some applications, such as vehicle sensor networks, require delays on the order of a few milliseconds.

To address the above concerns and push the limits of the BLE backscatter, we seek to answer a key question: *Can we establish fully compatible and authentic BLE connections between passive backscatter tags and commodity BLE devices with performance comparable to active BLE chips?* Upon delving deeper into this problem, we observe that enabling commodity BLE connections requires hardware resources that exceed the capabilities of existing backscatter systems. This

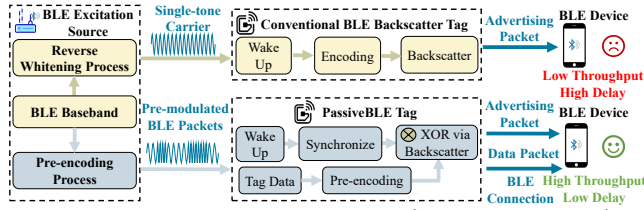


Figure 1: A PassiveBLE tag achieves commodity-compatible BLE connections with both data and advertising packets. In contrast, the traditional commodity-compatible tag [44] only employs BLE advertising packets to realize one-shot communication.

fundamental paradox brings the following three challenges in designing a fully commodity-compatible BLE backscatter.

- **Symbol-level synchronization.** Supporting commodity-compatible BLE connections demands μs -level latency (about $150 \mu\text{s}$) [35] and symbol-level accuracy (jitters $\leq 1 \mu\text{s}$) to generate data packets. The packet length detection approaches, which are widely used in BLE backscatter systems [17, 44–46], cannot meet the requirement due to the short packet interval in the BLE protocol [35] and insufficient accuracy (e.g., $2 \mu\text{s}$ [44–46]) due to envelope variances caused by fading of the wireless channel [28].
- **Tag-affordable BLE channel coding.** Two connected BLE devices need negotiation to acquire dynamic parameters e.g., CRC and data whitening parameters, for channel coding [35], which is unaffordable for backscatter tags. This is because the limited resources on battery-free tags and lack of local oscillators (LO) make it impractical to extract these parameters and encode data with these parameters within several microseconds.
- **Compatible BLE connection.** Establishing and maintaining a commodity BLE connection incurs significant storage and processing burdens on parameters negotiation (e.g., channel map exchange) and connection management (e.g., connection establishment, maintenance, and termination). Due to their limited processing capabilities, these power-intensive burdens are unaffordable for tags.

To tackle the first challenge, we propose to design a high-accuracy and low-latency synchronization circuit. Unlike conventional backscatter systems that detect the signal envelope [17, 44–46] or re-shaped envelope [11, 13, 25, 38] for synchronization, we extract the frequency difference between BLE symbols to perform synchronization. Since the frequency of a BLE signal has less variance than amplitude after propagation through a wireless channel [28], frequency difference extraction-based synchronization can achieve higher accuracy than envelope detection-based methods. Furthermore, since frequency difference extraction-based synchronization is realized by detecting the frequency difference of adjacent symbols, rather than detecting the length of the

whole packet [44–46], thereby significantly reducing the synchronization latency. The challenge is that backscatter tags have no LO to extract the symbol frequency difference. To address this issue, we leverage the fact that acoustic signals transmit much slower than electromagnetic (EM) signals, and thus use surface acoustic wave (SAW) filters to delay the symbols, and then mix them with non-delayed symbols. In this way, the frequency difference of neighboring symbols is converted to the baseband without LO.

To overcome the second challenge, we propose a distributed coding scheme that directly uses the RF signals for logic computing. We observe that the standard BLE systems use a series of operations including exclusive OR (XOR) logical operations to combine the dynamic parameters with the data for encoding. Particularly, we offload part of the channel coding operations that require dynamic parameters, which can only be acquired by fully decoding incoming packets, to the excitation source. To this end, we reformulate the operations according to the distributive and associative laws, in order to separate operations involving dynamic parameters and other operations as two inputs of the XOR operation. Specifically, the excitation source generates pre-modulated carriers with these parameters, and then the tags use the “XOR” operation to generate BLE data packets via scattering the pre-modulated carriers. At a high level, PassiveBLE’s design incorporates an RF computing mechanism that operates RF signals’ property for computing purposes. In this way, the backscatter tags do not need to decode the incoming BLE packets to acquire parameters for channel coding.

To solve the last challenge, we design a connection scheduler to offload the tasks of negotiation and connection management from tags to the excitation source. To relieve tags from on-board storage and process burdens, we modify the firmware of a commodity BLE transceiver as an excitation source to establish, maintain and terminate connections between tags and unmodified BLE devices. To further reduce the energy cost of tags from frequency hopping, the excitation source generates frequency-hopped carriers to achieve the frequency hopping spread spectrum (FHSS). Thus, resource-limited tags can conserve more energy for synchronization and data transmission tasks while retaining high-performance connections with commodity BLE devices.

With all the challenges addressed, we implement the PassiveBLE, as illustrated in Figure 1. The PassiveBLE tag achieves onboard BLE modulation with $491 \mu\text{W}$ energy consumption using off-the-shelf components. We also design an ASIC sketch using Synopsis Design Compiler [41] with a power consumption of about $9.9 \mu\text{W}$. Then, we conduct extensive experiments to evaluate the performance of PassiveBLE under both line-of-sight (LoS) and non-line-of-sight (NLoS) conditions. The results demonstrate that PassiveBLE achieves a success rate of over 99.9% in establishing

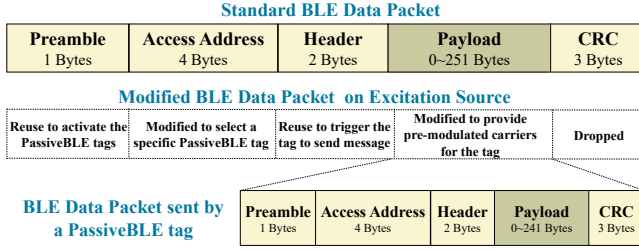


Figure 2: Reuse the structure of BLE data packets to achieve commodity-compatible.

commodity BLE connections. PassiveBLE achieves a goodput of up to 532 kbps in *LE 1M PHY* mode, which is about $63.3\times$ higher than those achieved by the state-of-the-art commodity-level BLE backscatter system [44]. PassiveBLE also achieves a goodput of up to 974 kbps in LoS conditions and up to 966 kbps in NLoS conditions in *LE 2M PHY* mode.

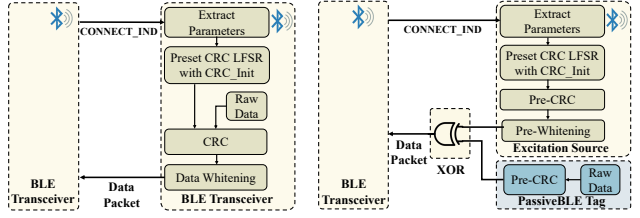
To the best of our knowledge, PassiveBLE is the first attempt to generate standard BLE packets using backscatter to enable fully compatible BLE connections. Main contributions are summarized as follows.

- We provide a low-power coherent-detection-based synchronization circuit for backscatter systems, which provides higher accuracy and lower delay than existing envelope-detection-based approaches. We achieve a sensitivity of up to -33 dBm via off-the-shelf components, with a synchronization accuracy of about 93 ns.
- We provide a distributed coding scheme to construct an XOR operation on RF signals, which allows backscatter tags to generate standard BLE data packets without knowing dynamic parameters.
- We provide a BLE connection scheduler to manage standard BLE connections between tags and commodity BLE devices. It allows backscatter tags to establish and maintain BLE connections with BLE devices such as smartphones without any modification.

2 Observation

Upon delving into the BLE communication schemes, we find the following two observations to realize BLE connections on backscatter tags.

First, we observe that the structure of BLE data packets provides an opportunity for the BLE excitation source to wake up the backscatter tags and generate BLE data packets with only software modifications. Figure 2 illustrates the packet structure of an *LE 1M* data packet. The *LE 2M* data packet follows a similar structure but features a 2-byte preamble. The preamble and header of the BLE packet can be reused to wake up and synchronize the backscatter tag without modification. With software-level modifications, the Access Address can be used to select a specific backscatter tag. The variable-length payload (0-251 Bytes) can be used



(a) Traditional BLE Connection. (b) BLE Connection on PassiveBLE.

Figure 3: Architecture of active BLE and PassiveBLE.

to generate a shorter BLE data packet with a variable-length payload (0-241 Bytes). Therefore, based on this observation, the activation of backscatter tags as well as BLE data packet generation, can be completed in a short period, which makes the BLE connection on backscatter tags possible.

Second, we observe that the standard BLE systems use XOR logical operations to mix the dynamic parameters with the data for channel coding, e.g., CRC generation and whitening, before modulating the data on an RF carrier [31]. Thus, we can decompose the encoding process into two parts: one on the excitation source to perform pre-modulation and the other on the backscatter tag to pre-process the data to be sent. Then, we can combine these two parts with an external XOR operation to generate commodity-level BLE data packets. Meanwhile, we observe that the toggling of the wireless signals' phase can be used to construct an XOR operation during the scattering process. Based on these observations, we intend to develop a distributive encoding scheme to generate standard BLE data packets on backscatter tags. The comparison with conventional BLE data packet generation is illustrated in Figure 3. We utilize the BLE excitation source to store the dynamic parameters and use these parameters to generate pre-modulated carriers for the backscatter tag. The tag combines the data with the pre-modulated carriers using the XOR operation to generate standard data packets.

3 Design

3.1 Design Overview

As shown in Figure 4, the PassiveBLE system consists of tags, an excitation source, and unmodified BLE devices. The excitation source is a commodity BLE transceiver that emits pre-modulated BLE data packets as carriers for backscatter tags. The PassiveBLE tag XOR data on the pre-modulated BLE packets to generate standard BLE packets with data payload. Then, the unmodified BLE device can directly decode the standard BLE packet to extract data from the tag. The excitation source can help one or multiple tags establish, maintain, and terminate standard BLE connections between tags and unmodified BLE devices. PassiveBLE employs the following hardware and software key designs to achieve BLE connections between tags and commodity BLE devices.

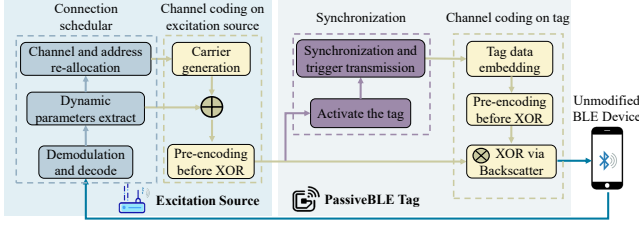


Figure 4: Overview of PassiveBLE system.

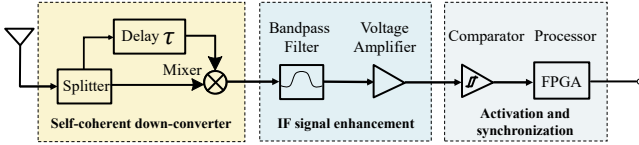


Figure 5: The high-level synchronization circuit schematic of PassiveBLE.

Synchronization. The synchronization circuit is implemented on the PassiveBLE tag. We propose a high-accuracy, low-latency synchronization circuit to enable μs -level latency in wake-up and synchronization of a selected PassiveBLE tag and ns-level accuracy to achieve symbol-to-symbol alignment between the excitation source and the tag. This design lays the basis to support the symbol-aligned XOR operation for encoding and quickly responding to commodity BLE devices during the connection.

Channel coding. Our key insight is directly using an RF computing mechanism to achieve lightweight channel coding. The channel coding scheme is implemented on both the PassiveBLE tag and the excitation source. It reduces the burden of encoding and processing operations on tags with dynamic parameters to achieve commodity-compatible high-rate encoding with lower power consumption. This design is the key to generating BLE data packets on resource-limited backscatter tags and achieving physical-layer compatibility with commodity BLE devices.

Compatible connection. The connection scheduler is implemented on the excitation source. It employs the excitation source to conduct all the required control and interactions, e.g., FHSS, establish, maintain and terminate the BLE connections between the backscatter tags and unmodified commodity BLE devices.

3.2 Synchronization

The high-level synchronization circuit is illustrated in Figure 5. It consists of three major components including (i) a self-coherent down-converter to extract the frequency difference of neighboring BLE symbols, (ii) an IF signal enhancement module that consists of a filter and an amplifier circuit to strengthen the extracted difference, (iii) an activation and synchronization module which consists of a voltage comparator and logical correlation circuits in FPGA to detect the preamble of BLE packets for synchronization.

Self-coherent down-converter. BLE signals are modulated with Gaussian Frequency Shift Keying (GFSK), which contains two types of symbols at different frequencies. Figure 6 shows a piece of BLE signal at center frequency f_c . The symbol “1” is represented with electromagnetic waves at frequency $f_1 = f_c + \Delta f/2$ and “0” at frequency $f_2 = f_c - \Delta f/2$. Upon reception of the signal at the down-converter circuit, the signal splits into two paths and one of the paths will add a ΔT time delay. Then we mix the delayed signal and the non-delayed signal. If neighboring symbols differ, it will produce an IF signal with a frequency $\Delta f = |f_1 - f_2|$, which can be used for synchronization after we filter it out.

Delay RF signals. Delaying the RF signals is challenging in a size-limited backscatter tag. Existing backscatter designs [48] usually use microstrip lines to delay signals. A 64 mm microstrip line can only delay the signal for 1.26 ns [27], which is insufficient to delay BLE symbols (≥ 500 ns) on a size-limited backscatter tag. To cope with this issue, we use SAW filter, which is commonly used to attenuate unwanted signals with different frequencies in backscatter systems [11, 17] or against metallic environments and interference in RFID systems [40], to delay signals. The principle of signal delay in a SAW filter is illustrated in Figure 7. One of the split signals passes through a short microstrip line and the other passes through a SAW filter. The signal passing through the microstrip line incurs negligible delay due to the high speed of EM signals (close to the lightspeed in vacuum [1]). The signal that passes through the SAW filter is converted to mechanical signals by an interdigitated transducer (IDT), and then propagates as mechanical signals in the SAW from one side to the other. After reaching the other side, the mechanical signals will be converted back to EM signals. As the mechanical signals are much slower (about 5.5 km/s [26]) than the EM signals, a small SAW filter will create a significant delay difference between these two signals. After testing various off-the-shelf SAW filters, we choose model EPCOS B8328 [10]. Three cascaded small SAW filters ($1.4 \times 1.1 \text{ mm}^2$) can provide about 46 ns time delay with about 3.3 dB insertion loss (measured by a Keysight MXR608A oscilloscope [20]). A 3.3 dB amplitude difference between mixed signals does not directly alter the output frequency accuracy, as frequency is determined by phase relationships. Then we build a single diode mixer to combine these two signals. It will output the IF signal with a frequency Δf , along with the linear combinations of original frequencies f_1 and f_2 , including $f_1 + f_2$, $2f_1$, $2f_2$, etc.

IF signal enhancement. To improve the performance of synchronization, we strengthen the IF signals at frequency Δf by filtering out unwanted signal and amplifying the IF signals. First, we design a low-pass filter with 3 dB bandwidth to 10 MHz to remove unwanted original signals, and their linear combinations. Then we implement a high-impedance

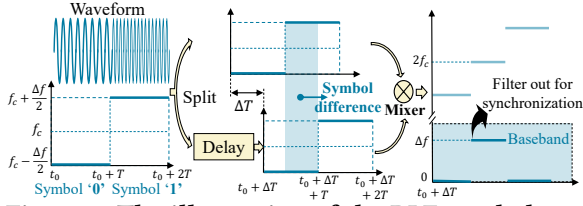


Figure 6: The illustration of the BLE symbol difference extraction.

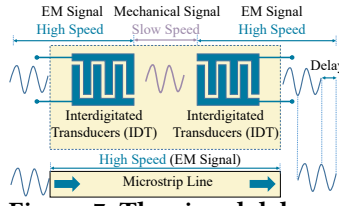


Figure 7: The signal delay effect of SAW filter.

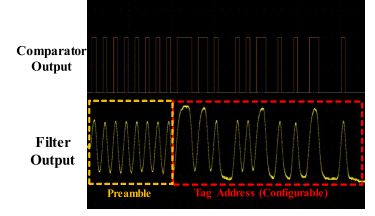


Figure 8: Output showcase.

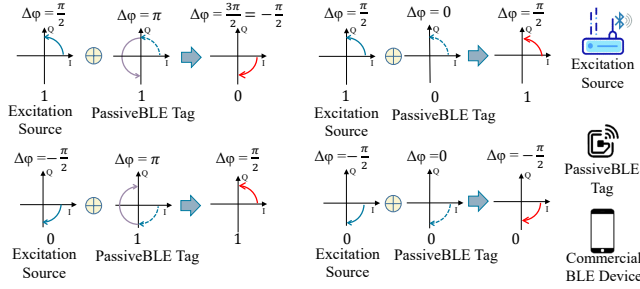


Figure 9: XOR operation via backscatter.

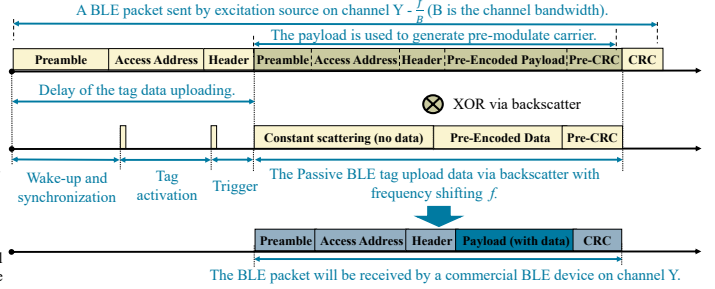


Figure 10: BLE packet generation in a PassiveBLE system.

voltage amplification circuit, which provides about 30 dB gain with μW -level power to amplify IF signals.

Activation and synchronization. The ability to extract the difference between symbols provides an opportunity to reuse the structure of BLE packets for tag activation and synchronization. Therefore, the preamble of the BLE packets, which is used for synchronization in commodity BLE devices, can also be used to wake up and synchronize the tags. The Access Address of the BLE packets, which is used to identify different BLE connections, is leveraged to identify different tags and support multiple BLE connections with tags in PassiveBLE. Meanwhile, as the Access Address is configurable, we can also re-allocate the Access Address with continuous “00” to stand for a bit “0” and “01” stands for a bit “1” to improve the sensitivity. As a tradeoff, the maximum number of connections that PassiveBLE can support will be reduced. For example, the 4 Bytes Access Address can be reused to allocate to $2^{32/n}$ tags if we use n different BLE symbols to stand for a bit “1”. As shown in Figure 8, we present a circuit output captured using a Tektronix MSO64B oscilloscope, clearly demonstrating that the preamble and tag address from an LE 2M packet can be successfully extracted.

3.3 XOR Operations on RF Signals

To overcome the challenge that tags do not have LO and insufficient processing capability to demodulate and decode BLE packets to extract the dynamic parameters, we observe in § 2 that the standard BLE channel coding process can be achieved by XOR-ing dynamic parameters at the excitation source with data on a tag. Thus, we intend to construct XOR operations directly on RF signals, which is called RF Computing [25], to build standard BLE packets with backscatter.

On one hand, commodity BLE transceivers decode the BLE symbols by estimating the phase accumulation $\Delta\phi$ during the symbols, where $\Delta\phi = \frac{\pi}{2}$ represents bit “1” and $\Delta\phi = -\frac{\pi}{2}$ represents bit “0”. On the other hand, a tag shifts the BLE signals with a square wave to another channel and modulates bits by changing the phase of the square wave [44]. Thus, by taking the phase accumulations caused by the excitation source and the phase change caused by the tag as input, and taking the phase demodulated by the unmodified BLE device as output, an XOR operation is constructed.

Figure 9 illustrates how XOR operations are constructed in PassiveBLE. When the excitation source provides bit “1” by generating a symbol with $\frac{\pi}{2}$ phase accumulation and the PassiveBLE tag provides bit “0” with a constant frequency shift with phase accumulation 0, the commodity BLE receiver will decode a bit “1” since the overall phase accumulation is $\frac{\pi}{2}$. The other three input cases are also consistent with the logic law of XOR operations. Consequently, the results in the receiver only contain two states and meet the logical law of XOR operations.

3.4 BLE Packets Generation with XOR Operations

Figure 10 explains how the XOR logical operation can be used to achieve BLE-compatible CRC generation and data whitening. After connection establishment, the unmodified BLE device listens on channel Y for data packet reception. Then, the excitation source will send a BLE packet to activate the tag and provide a carrier for the tag to transmit data on channel $Y - f_s/B$, where f_s is the frequency shift of the tag and $B = 2 \text{ MHz}$ is the bandwidth of a BLE channel. The Preamble, Access Address, and Header of the packet are used to activate

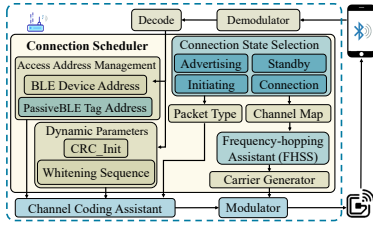


Figure 11: Connection scheduler structure.

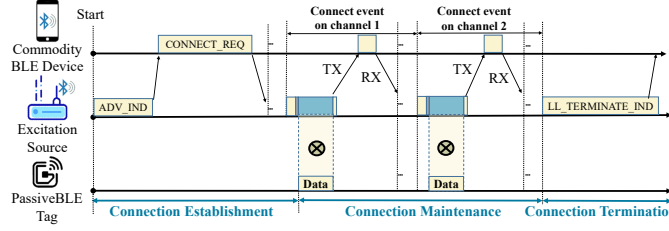


Figure 12: A connection lifetime example for a PassiveBLE tag and a commodity BLE device.

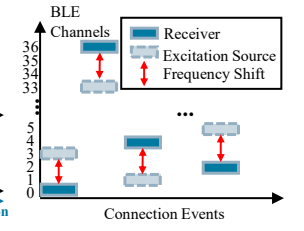


Figure 13: FHSS on tags.

the tag. We reallocate the first several bytes (seven bytes in LE 1M PHY or eight bytes in LE 2M PHY) of the original payload as the Preamble, Access Address, and Header of a new BLE packet sent by tags. These bytes are encoded by the excitation source and frequency-shifted by the tag to channel Y without phase changes. Rest of the payload is used by the tag as the new payload. Then, the tag encodes the message using CRC generation and data whitening through XOR operations, which are explained below.

CRC generation. After a BLE connection is established, dynamic parameters including *CRC_Init* and *Channel Index*, are determined. *CRC_Init* is the initial value of the CRC generator, and *Channel Index* is the index of the current channel of the BLE connection. When the BLE connection hops into a new channel, *Channel Index* changes.

In standard BLE systems, a transceiver obtains the dynamic parameter *CRC_Init* by decoding a received packet and then uses *CRC_Init* to generate a CRC checksum, which is used in the following packet the transceiver transmits. However, backscatter tags cannot decode the BLE packets due to the lack of LO and insufficient processing capability.

To address this issue, PassiveBLE tags are designed to generate the correct CRC checksum without knowing the *CRC_Init*. Given the message to be sent by the tag $m(x)$ and the initial state *CRC_Init* of the CRC generator, which is a 24-bit LFSR circuit that can be expressed as $g(x) = x^{24} + x^{10} + x^9 + x^6 + x^4 + x^3 + x + 1$, the CRC checksum *CRC_Value* according to the BLE specification can be computed as [31]

$$CRC_Value = (m(x) \oplus CRC_Init) \% g(x) \quad (1)$$

where $\oplus$ is the XOR operation and $\%$ is the modulo operation. Note that the modulo operation is implemented in the circuit using a series of modulo 2 subtractions, which satisfies the distributive law. Therefore, Eq. (1) can be expressed as

$$CRC_Value = (m(x) \% g(x)) \oplus (CRC_Init \% g(x)) \quad (2)$$

Thus, the tag modulates $(m(x) \% g(x))$ and the excitation source modulates $(CRC_Init \% g(x))$. Then, the tag and the excitation source generate *CRC_Value* by performing the XOR operation. By this means, without knowing the *CRC_Init* bits, PassiveBLE tag can also generate correct *CRC_Value* over the air and create standard BLE packets with the assistance of the excitation source.

Data whitening. Data whitening is used in BLE to enhance the stability of wireless links by avoiding long sequences of zeros or ones. Before transmitting packets, both the header and the payload undergo scrambling with a data whitening sequence to randomize bit sequences, reduce highly redundant patterns, and minimize DC bias. In BLE systems, the BLE signal $S_W(x)$ after whitening and modulation can be expressed as

$$S_W(x) = (m(x) \oplus w(x)) \times e^{2\pi f_c t} \quad (3)$$

where $m(x)$ is the message to be sent. $w(x)$ represents the whitening sequence and is derived from the current *Channel Index* [31]. $e^{2\pi f_c t}$ is the carrier signal. Since XOR operation satisfies the associative law, $S_W(t)$ can also be derived by

$$S_W(x) = m(x) \oplus (w(x) \times e^{2\pi f_c t}) \quad (4)$$

Thus, the excitation source modulates $(w(x) \times e^{2\pi f_c t})$. Overall, the pre-modulate process can be briefly concluded with the following equation:

$$E_{seq} = CRC_Init \% g(x) \oplus w(x) \quad (5)$$

where the $CRC_Init \% g(x)$ is the pre-processed CRC results and $w(x)$ is the whitening sequence for the data whitening process. The configuration and data loading process has a comparable level of complexity to the previous BLE backscatter systems (e.g., InterScatter [16], IBLE [44], etc.), which also load pre-modulated data in the payload part of BLE advertising packets to generate single-tone carriers. The tag only needs to modulate $(m(x) \% g(x))$ and perform the XOR operation according to Eq.(2) and Eq.(4).

3.5 PassiveBLE Connection Scheduler

So far we have accomplished BLE data packet construction on a PassiveBLE tag. The final step towards achieving standard-compatible BLE connections is to implement the commands and interactions for connection establishment, maintenance, and termination. To achieve this goal, we design a connection management scheduler on the excitation source to assist PassiveBLE tags to establish, maintain, and terminate connections with unmodified commodity BLE devices. The structure of the scheduler is illustrated in Figure 11, where our modifications are highlighted with light blue and the reused BLE processing modules are in yellow. We employ three new components to enable the connection scheduler,

including (i) a connection state management module to determine the proper packet to be sent in different connection states, (ii) a frequency-hopping assistant to help the PassiveBLE tags accomplish the FHSS mechanism according to BLE specifications, and (iii) an access address management module to reallocate addresses for multiple PassiveBLE tags.

Connection state management. The connection state selection module assists PassiveBLE tags during the whole connection lifetime, which consists of four basic states, including advertising, initiating, connection, and standby states. In the advertising state, the scheduler can help tags send and receive advertising packets. In the initiation stage, the scheduler helps the PassiveBLE tag negotiate with the active BLE devices and receive the dynamic parameters for the connections. In the connection state, the scheduler extracts the channel map to assist the frequency hopping during different connection events. In the standby mode, the scheduler helps tags terminate BLE connections. Figure 12 illustrates a connection lifetime example. First, an excitation source sends an advertising packet, which can also be sent by a tag, to request a BLE connection with the tag. Then, the commodity BLE device answers the request to initiate connection establishment between the tag and the BLE device. Next, the tag XORs the data on the BLE data packets generated by the excitation source to generate standard BLE packets for commodity BLE devices. After several connection events, the excitation source, or the BLE device, can terminate the connection by sending a connection termination control packet.

Frequency-hopping assistant. We employ the excitation source to help tags perform frequency hopping by generating carriers at different channels due to the following reasons: (i) FHSS requires the *Channel Index* which cannot be obtained by tags, and (ii) synthesizing multiple frequencies is difficult and power-hungry for an energy-limited PassiveBLE tag. As illustrated in Figure 13, we use the excitation source to extract the frequency hopping pattern from the *CONNECT_IND* packet and provide a carrier with a dynamic frequency $f_t - f_s$, where f_t is the desired frequency according to FHSS at time t . The PassiveBLE tag modulates signals with a fixed frequency shift f_s . Consequently, the PassiveBLE tag only needs to transmit the data upon being triggered without knowing the frequency hopping pattern. The frequency hopping operation of the PassiveBLE system can be briefly concluded into the following 3 steps. First, the excitation source takes the channel hopping map and negotiates with commodity BLE devices to determine the channel for connection dynamically. Then, the excitation source generates the pre-modulated packets on a channel that is f_s (frequency shift by the PassiveBLE tag) away from the target channel. Finally, the tag shifts the frequency and generates commodity-compatible packets on the target channel. The

synchronization of the tags is based on the preamble of the BLE data packets. Since the frequency hopping is determined by the excitation source, the PassiveBLE tags only need to detect the preamble and access address of the data packets, if the tag detects the correct preamble and access address, it will directly modulate the data on the payload part of the packets. The frequency hopping will be completed with the excitation source hopping in different channels.

Access Address management. When a connection is established, each BLE device is allocated a unique Access Address. To allow PassiveBLE tags to communicate with unmodified commodity BLE devices, the excitation source stores the Access Address for each tag and adds it to the header of the PassiveBLE packets. When there are multiple connections with different PassiveBLE tags, the excitation source manages these connections through time divisions.

4 Implementation

4.1 BLE Transceivers

A PassiveBLE system contains a BLE excitation source to generate carriers for PassiveBLE tags. We use a BLE development kit, model Nordic nRF52833 DK [32] to implement the excitation source. The excitation source supports both PHY modes defined in BLE, i.e., the *LE_1M_PHY* mode that is compatible with all existing BLE devices, and the *LE_2M_PHY* mode that is introduced by BLE protocol stack version 5.0. The Equivalent Isotropic Radiated Power (EIRP) of the excitation source is set to 20 dBm, which is compatible with the frequency regulation for BLE devices. These operations to configure a BLE transceiver as an excitation source can be easily achieved with the open API of the Nordic BLE board. A detailed tutorial can be accessed at the product's website [33]. The BLE devices, such as the receiver, can be any BLE device without firmware or hardware modification, such as unmodified smartphones.

4.2 PassiveBLE Tags

COTS implementation. As shown in Figure 14, we implement PassiveBLE prototypes with off-the-shelf components. A PassiveBLE tag consists of three major parts, a synchronization circuit to activate and trigger the tag for communication, an RF switch-based backscatter circuit for uplink communication and an ultra-low-power FPGA (IGLOO AGLN060 [24]) for baseband processing. The synchronization circuit contains the following passive lumped components, including an impedance matching network and a bandpass filter implemented with inductors and capacitors, three cascaded SAW filters (Qualcomm B39242B8328P810 [10]) to delay the signal, a single-diode mixer implemented with a splitter, a combiner implemented with microstrip lines, and a Schottky diode (Infineon BAT63 [15]), which can be used to build single-diode mixer with -55 dBm sensitivity [30] in 2.4 GHz frequency

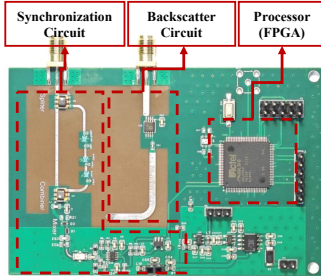


Figure 14: PassiveBLE tag.

band. We also use two low-power baseband amplifiers (Infineon LTC6262 [7]) and a comparator (ST TS3201 [39]) to convert the signal to binary bits for synchronization. The backscatter circuit comprises an RF switch (ADI ADG902 [6]) with microstrip lines.

Power Consumption. All of the active components are powered with a 1.8 V DC supply. There are two modes for the PassiveBLE tag, i.e., uplink communication mode and standby mode. In the uplink communication mode, the tag will upload data via scattering and the synchronization circuit will be disabled. The overall power consumption in this mode is about 491 μW , including 489 μW dynamic power consumption of the FPGA and 2 μW cost by the RF switch. The PassiveBLE works in the standby mode waiting for activation after the transmission. In this mode, overall power consumption is about 1.03 mW, including 35 μW consumed by the FPGA, 864 μW consumed by the amplifiers, and 131 μW consumed by the comparator. In active mode, the energy consumption of a PassiveBLE tag with both modes enabled is about 1.5 mW, which is much lower than the existing BLE backscatter system and BLE chips. But current COTS design cannot support low-power standby mode due to the poor static leak current of the off-the-shelf components and the full-duty-cycled synchronization circuit. Engineering efforts from these two aspects will further reduce the power level, and we believe it will perform better than BLE transceivers after being integrated into IC chips with less leakage current.

ASIC design. To explore the potential power consumption reduction performance, We also conduct an IC design using Synopsys Design Compiler [41] with TSMC 90 nm CMOS

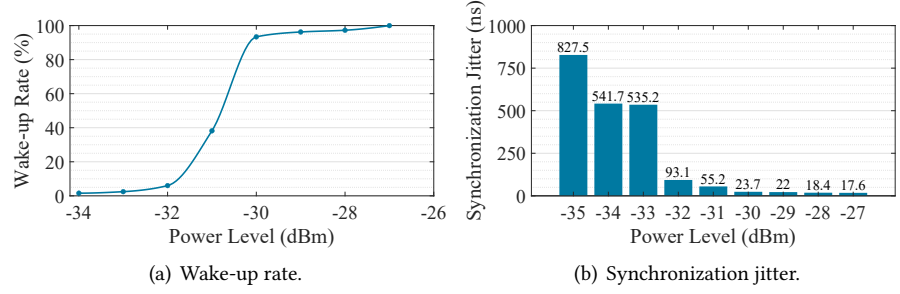


Figure 15: Wake-up and synchronization performance.

process. We provide a Verilog description of the digital state machine for baseband processing, BLE packet encoding and frequency shift. Then use the standard cell to simulate the Application Specific Integrated Circuit (ASIC) of PassiveBLE tag. The static power consumption is 0.9 μW and the total dynamic power consumption for uplink is 9.9 μW . We have also simulated the power consumption of the wake-up and synchronization circuits, which consume 190.3 μW in total, where the amplifier consumes 162.4 μW and the comparator consumes 27.9 μW . In Table 1, we compare both COTS and ASIC designs of PassiveBLE tag with commodity BLE chips and existing BLE backscatter designs, including DanBlue [17] and IBLE [44], etc, to show the potential of PassiveBLE for commodity-level BLE communications. As shown in Table 1, we see that a PassiveBLE tag consumes less than 24.8% power to achieve 7-115 $\times$ goodput compared to existing BLE backscatter systems. Meanwhile, a PassiveBLE tag also consumes 10-100 $\times$ less power to achieve 72.8% throughput of commodity BLE chips [32].

BLE connection capability. We also list the measured goodput and connection capability of existing BLE backscatter systems in Table 1, there are two existing systems, i.e. DanBlue [17] and Biblue [18], can support the BLE connections. But their connections are not commodity-compatible, which needs to pre-allocate static coding parameters for the connections. In standard BLE protocols, as addressed in Section 2 in the manuscript, these parameters are dynamically allocated in commodity BLE devices instead of pre-allocating. Unlike these systems, our system is much more commodity-compatible since it can support the BLE connection with dynamic parameters.

5 Evaluation

5.1 Synchronization Performance

Synchronization performance is crucial to support the symbol-to-symbol XOR operation and quick response to maintain the BLE connections. Here we verify the tag's synchronization performance with micro-benchmark experiments to show the wake-up, activation rate and synchronization performance. We use two metrics to evaluate the tag's

Table 1: Performance of existing BLE backscatter.

Systems	Power	Goodput	Connection
PassiveBLE	491 μW 9.9 μW (IC)	974 kbps	✓
DanBlue [17]	825.4 μW (IC)	30 kbps	✓
BiBlue [18]	240.9 mW	17 kbps	✓
IBL [44]	2 mW	8.4 kbps	×
Bitalign [13]	1.9 mW	131.7 kbps	×
RBLE [45]	37 μW (IC)	16.6 kbps	×
FreeRider [46]	30 μW (IC)	57 kbps	×

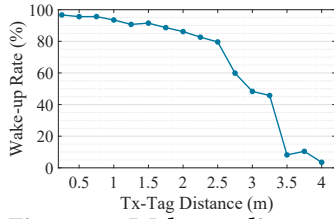


Figure 16: Wake-up distance.

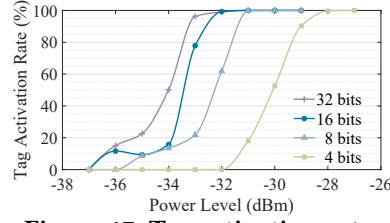


Figure 17: Tag activation rate.

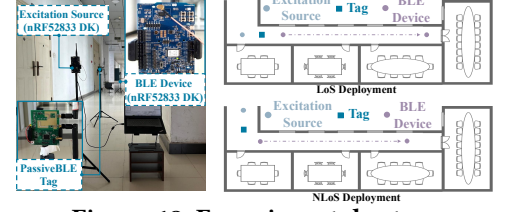
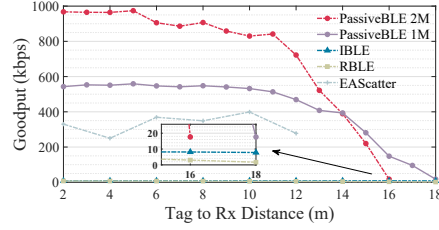
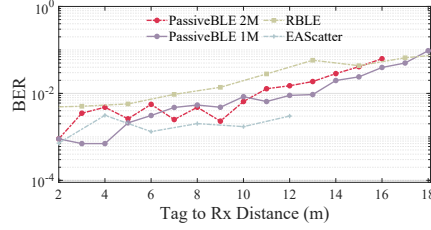


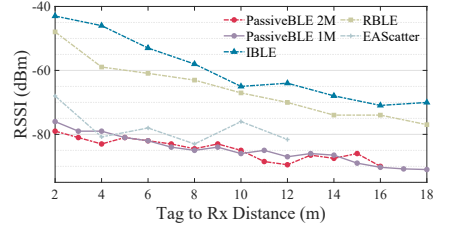
Figure 18: Experimental setup.



(a) Goodput.



(b) BER.



(c) RSSI.

Figure 19: End-to-end communication performance with increasing distance in the LoS condition.

sensitivity: wake-up rate and activation rate. The wake-up rate represents the tag's ability to achieve symbol-level synchronization with the excitation source's data packet. The activation rate refers to the tag's capability to detect its own ID address from the pre-modulated BLE packets generated by the excitation source. Additionally, we define synchronization jitter as the accuracy of symbol alignment between the backscatter tag and the excitation source.

Wake-up sensitivity. To evaluate the performance of the sensitivity to wake up a PassiveBLE tag, we configure a BLE transceiver as the excitation source to send BLE data packets with power levels from -40 dBm to -25 dBm. The excitation source and the tag are connected with a cable to reduce the varying wireless channel's influence on the results. If the tag has been woken up by the data packet, it will send a trigger signal. At each power level, we send 10,000 data packets and calculate the wake-up rate by dividing the count of trigger signals by the total packets sent. The results are plotted in Figure 15(a). The sensitivity is about -30 dBm where the successful rate is about 91%. When the power level falls below -31 dBm, the successful rate rapid drops to 24%.

Tag activation rate. As stated in § 3.2, we can use redundant bits in the *Access Address* part to improve the sensitivity of tag activation. We conduct a micro-benchmark experiment with the same setup to evaluate the tag activation rate with different bits. As plotted in Figure 17, the more redundant bits lead to a better tag activation rate with low power level BLE data packets. Since the wake-up and synchronization sensitivity with the BLE preamble is about -30 dBm, we choose $n = 8$ to balance the wake-up sensitivity and the number of tags, where $2^{32/8} = 16$ tags are sufficient for some BLE backscatter applications in wearable or smart homes.

Synchronization jitter. To quantify synchronization accuracy, we use synchronization jitter, as is standard in existing backscatter synchronization systems [9]. Jitter represents the time difference or errors between the symbol rising edges of the excitation source and the tag. We use a setup similar to the previous system [9] to measure the time jitter but configure the excitation source to pull up the voltage of an onboard input-output (IO) interface to mark the beginning of the packet. The tag will also pull up the voltage of an IO interface after it successfully detects the preamble. We use an oscilloscope to monitor the delay of the voltage pull-up edge between these two IO ports to acquire the synchronization latency and record the jitter of the edge of the tag's IO interface. The average delay from the transceiver sending the data packets to the PassiveBLE tag detecting the signal is about $10.4 \mu\text{s}$, where the preamble lasts for $8 \mu\text{s}$ and the onboard processing costs $2.4 \mu\text{s}$. The jitter at different power levels is plotted in Figure 15(b). The jitter is below 23.7 ns with a power level above -30 dBm and 93.1 ns with a power level above -32 dBm, which achieves the desired optimal spec for ns-level synchronization accuracy.

Synchronization distance. We fix a BLE excitation source at the end of a corridor with EIRP of the excitation source at 20 dBm, then move a PassiveBLE tag along the corridor with increasing distances away from the excitation source from 0 m to 5 m with 0.25 m increments. We configure the excitation source to transmit a dedicated BLE packet with a standard preamble and the tag's address is loaded in the *ACCESS ADDRESS* part. At each distance, the excitation source transmits ten thousands BLE packets to trigger the PassiveBLE tags, and we record the times that successfully wake up and trigger the tag. As plotted in Figure 16, results show that the success rate keeps above 80.0% with

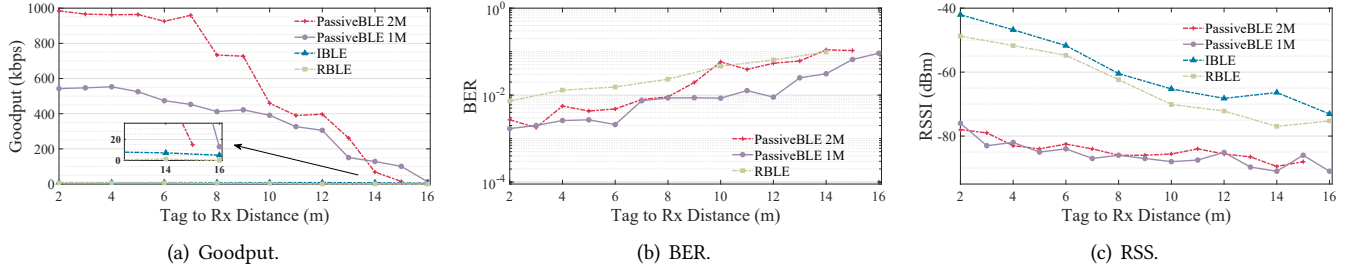


Figure 20: End-to-end communication performance with increasing distance in the NLoS condition.

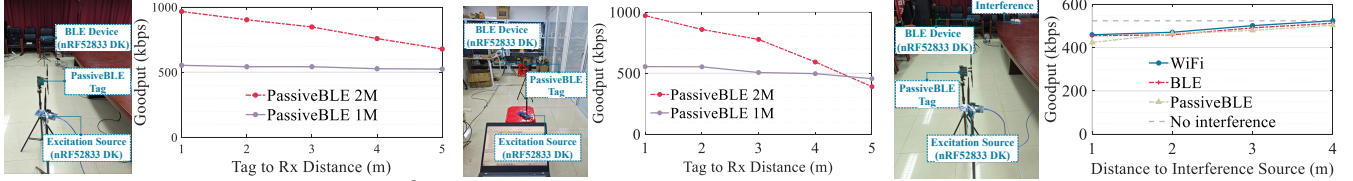


Figure 21: Communication performance in a meeting room.

Figure 22: Communication Performance in a cluttered lab.

Figure 23: Performance under different interference sources.

the distance increase to 2.5 meters and above 95.1% when the distance to the excitation source is within 1.5 meters.

5.2 Communication Performance

Experimental setup. As shown in Figure 18, we evaluate the performance of PassiveBLE in both line-of-sight (LoS) and non-line-of-sight (NLoS) scenarios. We use the TX-to-Tag distance to refer the distance from an excitation source to a PassiveBLE tag, and the Tag-to-RX distance to refer the distance from a PassiveBLE tag to the receiver. The default TX-to-Tag distance is set to 0.5 m, the same as the previous BLE backscatter system [44] for a fair comparison. In the LoS scenarios, we place the excitation source, tags, and receivers in the corridor, while in the NLoS scenarios, the excitation source and tags are placed at the corner of the corridor. We move the receiver along a straight line to increase the distance of Tag-to-RX distance and record the received packets for evaluation. In each location, we employ the BLE transceiver to receive packets for 5 minutes and calculate the average performance. Then compare our results with the SOTA solutions, including IBLE [44], EAScatter [14] and RBLE [45], with the published performance in their papers [14, 44]. In particular, IBLE and RBLE are compared in both LoS and NLoS scenarios and EAScatter [14] is compared in LoS. Some BLE backscatter systems [13, 17, 18] are excluded because of their different setup or lack of similar experiments, their measured goodput is listed in the Table 1.

We test the performance of PassiveBLE tags in both of the *LE 1M PHY* and *LE 2M PHY* modes in these two scenarios. The results are plotted in Figure 19 and Figure 20. Overall, PassiveBLE achieves remarkable performance improvement in goodput and bit error rate (BER) with a lower receive signal strength (RSS). PassiveBLE achieves full compatibility with unmodified BLE transceivers and reaches a

goodput of up to 532 kbps in *LE 1M PHY* mode, which is $63.3\times$ higher than that of the existing commodity-level BLE backscatter [44], in the LoS scenario. Although the goodput of the PassiveBLE tag drops quickly when the distance exceeds ten meters, PassiveBLE still realizes $3\times$ higher goodput than the commodity compatible IBLE system. Our experiment confirms that PassiveBLE outperforms the SOTA commodity-compatible backscatter system [44] in terms of goodput. We now elaborate on the performance in LoS and NLoS scenarios, respectively.

LoS scenario. As plotted in Figure 19, the performance of PassiveBLE is superior from both of the goodput and RSS aspects in most distances. In LoS scenarios, PassiveBLE achieves a high goodput up to 532 kbps in *LE 1M PHY* mode and 974 kbps in *LE 2M PHY* mode, which achieves orders of goodput improvement than existing BLE backscatter systems [17, 18, 44–46] and comparable to active BLE chips in most BLE applications [3, 34]. Furthermore, as plotted in Figure 19 (b), PassiveBLE presents lower RSS performance to achieve the higher goodput, which indicates the PassiveBLE is more power efficient. We also conduct experiments in two indoor scenarios, a meeting room and a cluttered lab, both of which tend to suffer from severe multipath interference, to evaluate the communication performance in more complex environments. Results are plotted in Figures 21 and 22. The goodput of PassiveBLE tags indeed degrades in these environments, with the degradation in *LE 2M* mode being more significant. Nonetheless, the system can still maintain about 500 kbps (in the meeting room) or 480 kbps (in the clutter lab) within 5 meters, thereby meeting the application requirements in these environments.

NLoS scenario. As plotted in Figure 20, PassiveBLE also achieves a superior goodput when the Tag-to-RX distance is

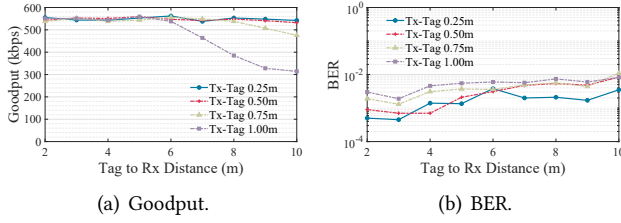


Figure 24: Performance versus different distances in LE 1M Mode.

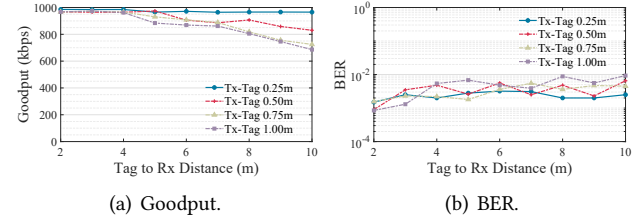


Figure 25: Performance versus different distances in LE 2M Mode.

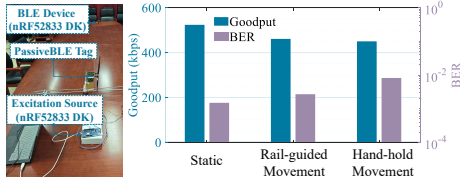


Figure 26: Performance under different movement.

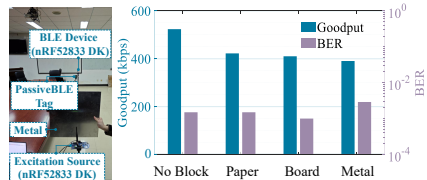


Figure 27: Performance with different blocks.

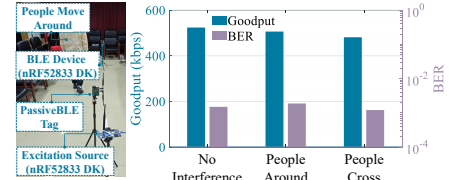


Figure 28: Performance under interference with nearby people.

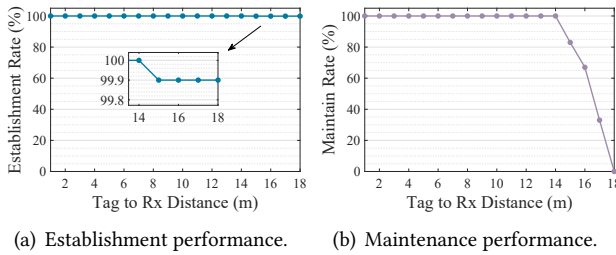


Figure 29: Connection performance.

below ten meters in NLoS scenarios. The measured average goodput of PassiveBLE tag with a Tag-to-RX distance of ten meters is 459.5 kbps in *LE 2M PHY* mode and 391.0 kbps in *LE 1M PHY*, which dropped as expected due to the occlusion. However, the performance of PassiveBLE is still sufficient for many IoT applications, e.g., sensor data collection [12] and audio streaming [36].

Impact of interference. We conduct an experiment to evaluate the performance of the PassiveBLE system under various interference sources, including a WiFi device (implemented with a USRP x310) transmitting OFDM symbols in the 2.4 GHz band, a PassiveBLE tag and BLE devices transmitting on random channels. As shown in Figure 23, the goodput drops by about 20% when the interference source is within 1 meter, yet it remains superior with ample room for performance enhancement through redundant coding methods. The influence is minimal when the distance is beyond 4 meters, indicating that this approach can meet the requirements of smart home or logistics applications.

Impact of Tx-to-Tag distance. The Tx-to-Tag distance is also a major concern when deploying PassiveBLE tags in practical applications. To evaluate how this distance influences the performance, we move the location of the tag

with the distance increasing from 0.25 m to 1.00 m and send ten thousand packets at each location in both BLE physical modes. The results of PassiveBLE in LE 1M and LE 2M modes are plotted in Figure 24 and Figure 25. Overall, when the Tx-to-Tag distance is increased to one meter, PassiveBLE can guarantee a goodput over 310 kbps and 680 kbps in LE 1M and LE 2M mode at the uplink distance of ten meters.

Reliability under dynamic environments. We evaluate performance in dynamic environments, including scenarios with different movements (Figure 26), blocking conditions (Figure 27), and movements of nearby people (Figure 28), with the Tag-to-RX distance fixed at 5 meters. Specifically, blocking the LoS between the tag and receiver causes the greatest performance degradation, followed by the effect of different movements, while the movement of nearby people has only a slight influence. Overall, even in the worst-case scenario, when the tag is partially blocked by a metal board, it can still provide a superior goodput of up to 390 kbps, offering attractive application potential for short-range, low-power wireless network applications.

5.3 Connection Performance

A significant advantage of PassiveBLE is that it is the first BLE backscatter system capable of establishing commodity BLE connections with unmodified BLE transceivers, rather than one-shot communication. In this subsection, we conduct experiments to evaluate the performance in establishing and maintaining the BLE connections.

Connection establishment performance. To evaluate the successful connection establishment rates with respect to distance changes, we configure the PassiveBLE tag to send advertising packets periodically, while the receiver listens and sends connection requests. We consider the connection

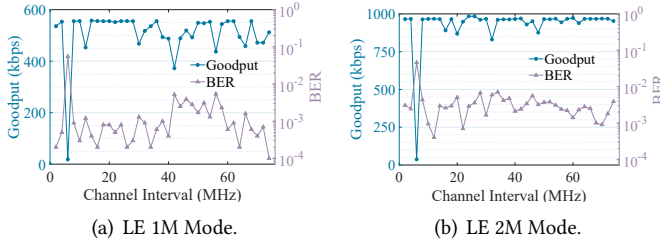


Figure 30: Impact of frequency shift between two PassiveBLE tags.

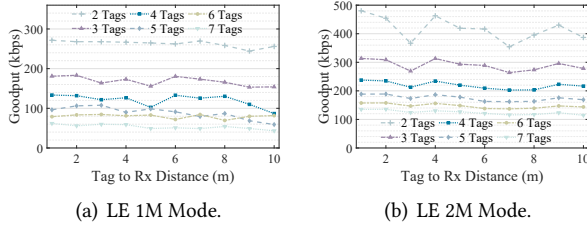


Figure 32: Average goodput of multiple tags with one excitation source.

to be established upon a successful handshake in the advertising channel and receipt of a data packet from the PassiveBLE tag by the receiver. In contrast, the connection fails if it cannot be established within 6 consecutive connection events, as per the BLE protocol [35]. The connection-establishing process is repeated over 1000 times at each location. Figure 29 (a) shows the experimental results, which confirm that PassiveBLE achieves a success rate of more than 99.9% in establishing BLE connections.

Connection maintenance performance. We also conduct experiments to evaluate the connection maintenance performance of PassiveBLE tags. We manually set a 10-second connection maintenance target, and the connection event is configured as 50 ms. Similarly, a maintenance failure occurs if 6 consecutive connection events are lost. We repeat the 10-second connection maintenance experiments over 100 times at each distance. As plotted in Figure 29 (b), PassiveBLE tag can achieve a successful connection maintenance rate at 100% within 14 meters. Then it degrades to below 33% when the distance exceeds 17 meters. The success rate of connection maintenance is lower than the connection establishment rate under the same distance, which meets expectations. Under the same packet loss rates, BLE connection establishment exhibits higher success probability because it only requires at least one successful handshake packet within six attempts. In contrast, connection maintenance involves prolonged data transmission, where the extended operational duration statistically increases exposure to rare six-packet loss streaks, despite equal per-packet failure probabilities. The connection maintenance faces a higher probabilistic risk due to longer exposure time to channel variability.

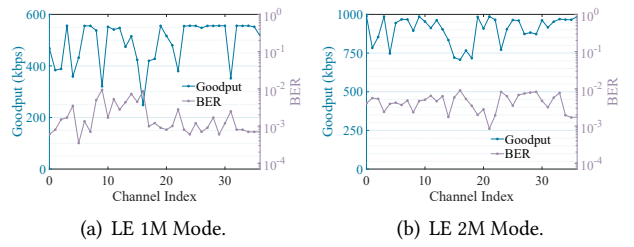


Figure 31: Performance of a PassiveBLE tag in different BLE channels.

Impact of the frequency hopping. Here we conduct a micro-benchmark experiment to evaluate the communication performance with/without the FHSS assistant on the excitation source. The PassiveBLE tag is 0.5 m away from the excitation source and the receiver is 5 m away from the tag. We first configure the excitation source to send in fixed channels. In each channel, we collect 1000 BLE data packets generated by the tag to calculate the goodput and BER in that channel. The results in all of the BLE data channels that we tested one by one are plotted in Figure 31. The goodput and BER fluctuate roughly in different channels, where the goodput in the worst channel is less than half of the goodput in a good channel, e.g., the 240 kbps in channel 17 and 540 kbps in channel 24 as shown in Figure 31(a). We then test the performance of FHSS with randomly selected channels. In each channel, we collect 30000 packets in both of the modes to calculate the goodput and BER. Overall, the goodput is 902 kbps and 490 kbps in LE 2M and LE 1M mode, which is 93% of the best case.

Impact of frequency shift. We also conduct experiments to evaluate the impact of interference between two BLE tags that have different frequency shifts. We configure an excitation source to generate carriers at channel 19. Then configure two synchronized tags to generate BLE data packets with the carriers. The frequency interval of these two tags is configured ranging from 0 MHz, i.e. the same channel, to 80 MHz, i.e. over the whole BLE channels. In each frequency interval, we generate 1000 BLE data packets and use two BLE transceivers to collect these packets and calculate the average goodput and BER in these frequency intervals. As plotted in Figure 30, the frequency interference between two tags with different frequency shifts, results in fluctuating goodput and BER. But overall, PassiveBLE achieves over 400 kbps goodput and $\leq 1\%$ BER, which suffices for typical BLE applications.

Impact of multiple tags on one excitation source. In this subsection, we evaluate the performance of PassiveBLE in maintaining multiple BLE connections with tags. As addressed in § 3, the BLE connection scheduler manages multiple connections using time division duplex methods [35].

To evaluate the performance of PassiveBLE in such scenarios, we conduct experiments with multiple PassiveBLE tags connected to one BLE transceiver with one excitation source, and the duration of each tag's connection is configured to be $N \times 50$ ms. We set 10 test locations with uplink distances ranging from 1 m to 10 m, and at each location, we record 5 minutes of data packets to evaluate the goodput. After all of the tags have successfully established the connection with the transceivers, the transceiver starts to collect the BLE packets and estimates the goodput. Results are plotted in Figure 32, the average goodput of each tag drops rapidly than expected, which may be caused by the processing delay. Overall, the average goodput for one tag can stay above 39 kbps, which is sufficient for logistic applications [46].

6 Related Work

BLE-based backscatter. As one of the most common ultra-low-power IoT communication methods, BLE backscatter [13, 14, 16–18, 29, 44–46] has garnered substantial attention in recent years. In particular, FreeRider [46] and InterScatter [16] open up new opportunities of ambient IoT networking by scattering widely-adopted Bluetooth signals for ultra-low-power communications. Delving further, RBLE [45], IBLE [17, 44], and EAScatter [13, 14] have dedicated substantial efforts to improve the reliability, compatibility, and goodput to enable practical BLE backscatter applications. Different from these systems, PassiveBLE takes the first step to generate standard-compatible BLE data packets and establish fully commodity-compatible BLE data connections with unmodified BLE devices such as smartphones.

Low power demodulator. Recently, the research community has shifted the focus of backscatter designs to new RF demodulators [9, 11, 30, 38] to replace the conventional envelope-detection-based demodulator. These demodulators are specifically designed for LoRa or WiFi signals, either relying on the frequency-increasing feature of chirp signals [11, 38] or requiring multiple WiFi subcarriers [30]. SyncScatter [9] designs and implements an outstanding wake-up radio chip for WiFi backscatter, which achieves 150 ns accuracy through a novel two-stage hierarchical envelope comparing methods. However, SyncScatter needs three WiFi packets (about $432\mu\text{s}$) to wake up tags, which is not suitable for our system due to the longer intervals of packets in the BLE protocol. Departure from these approaches, PassiveBLE demodulates the RF signals by comparing the signal with a delayed copy of itself, which meets the need for short delay and high accuracy in standard BLE communications.

Distributed Processing. Distributed processing technologies [19, 21, 23, 47] in the networking area, such as processor sharing [21, 23] of backscatter and network coding [19, 47], pioneer the practice of offloading the burden

from resource-limited devices to reduce power consumption or lower multicast traffic and improve network throughput. This insight inspires us to design a distributed coding scheme that offloads certain channel coding operations from tags to the excitation source. However, the XOR operation in PassiveBLE differs technically from those systems, in which distributed processing occurs in the baseband with logical circuits, while we construct XOR operations with RF signals.

7 Discussion

Communication Distance. The PassiveBLE system can achieve a maximum distance of about 17 meters with hundred-kbps-level goodput, which meets the requirements for short-range communication (e.g., wearable electronics, in-vehicle sensor networks, and logistics). The communication range still requires improvement compared to commodity active BLE chips, thereby limiting its usage in other BLE scenarios requiring longer ranges (e.g., device localization). The range is constrained primarily by the sensitivity (about -30 dBm) of the current synchronization circuit implementation. We believe that implementing the system with tunnel diodes [8, 43] or using μW level LNA IC designs [42] can further extend the communication distance.

BLE security. Using advertising packets for data upload-ing will expose the backscatter system to security risks, including eavesdropping, spoofing and data leakage, etc. Because the advertising packets are open to any BLE devices with publicly accessible rules. PassiveBLE system can enhance the security performance by leveraging BLE data packets for communication since the data packets are hard to eavesdrop with dynamic coding rules and seeds.

8 Conclusion

We have designed and validated PassiveBLE, which takes the first step to establish fully compatible and authentic BLE connections between passive backscatter tags and unmodified commodity BLE devices. We explore the design space of the low-power coherent-detection-based synchronization circuit, the XOR-based distributed coding scheme, as well as the excitation source-aided connection scheduler in backscatter communications. By extending backscatter communications to commodity BLE connections over data channels, PassiveBLE can accommodate a wide range of BLE-suitable use cases.

Acknowledgment

We thank the anonymous reviewers and our shepherd, Yichao Chen, for their valuable comments. This research is supported in part by the National Natural Science Foundation of China with Grant 62471194, 62425207, Research Funds for the Central Universities, RGC under Contract CERG 16204523, AoE/E-601/22-R.

References

- [1] Pavel Andreyev, Alexey Grishko, and Nikolay Yurkov. 2016. The temperature influence on the propagation characteristics of the signals in the printed conductors. In *2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*. IEEE, 376–378.
- [2] Roshan Ayyalasamayajula, Deepak Vasisht, and Dinesh Bharadia. 2018. BLoc: CSI-based accurate localization for BLE tags. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*. 126–138.
- [3] ABI research Bluetooth SIG. 2024. The Ambient IoT: The Emergence of a New Class of Bluetooth® IoT Devices. https://www.bluetooth.com/wp-content/uploads/2024/01/MRN_Ambient-IOT.pdf. Accessed: July 2024.
- [4] Ishan Chatterjee, Maruchi Kim, Vivek Jayaram, Shyamnath Gollakota, Ira Kemelmacher, Shwetak Patel, and Steven M Seitz. 2022. ClearBuds: Wireless binaural earbuds for learning-based speech enhancement. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*. 384–396.
- [5] Williot Company. 2022. Ambient IoT: What is it? How will it impact Industries and Consumers? <https://www.wiliot.com/ambient-iot>. Accessed: July 2024.
- [6] Analog Devices. 2017. ADG902 Datasheet. <https://www.analog.com/cn/products/adg902.html>. Accessed: July 2023.
- [7] Analog Devices. 2024. Operational Amplifier Components. <https://www.semiee.com/file2/0f6682c242f5dab7341f0d8ccb99fead/ADI/ADI-LTC6262.pdf>. Accessed: September 2024.
- [8] Huixin Dong, Yirong Xie, Xianan Zhang, Wei Wang, Xinyu Zhang, and Jianhua He. 2023. Gpsmirror: Expanding accurate gps positioning to shadowed and indoor regions with backscatter. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.
- [9] Manideep Dunna, Miao Meng, Po-Han Wang, Chi Zhang, Patrick Mercier, and Dinesh Bharadia. 2021. SyncScatter: Enabling WiFi like synchronization and range for WiFi backscatter Communication. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI '21)*. 923–937.
- [10] EPCOS. 2014. SAW Components. <https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/1111/B39242B8328P810.pdf>. Accessed: July 2024.
- [11] Xiuzhen Guo, Longfei Shangguan, Yuan He, Nan Jing, Jiacheng Zhang, Haotian Jiang, and Yunhao Liu. 2022. Saiyan: Design and implementation of a low-power demodulator for LoRa backscatter systems. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI '22)*. 437–451.
- [12] Jun Huang, Mingli Zhao, Yide Zhou, and Cong-Cong Xing. 2018. In-vehicle networking: Protocols, challenges, and solutions. *IEEE Network* 33, 1 (2018), 92–98.
- [13] Zhanxiang Huang, Yuan Ding, Dapeng Oliver Wu, Shuai Wang, and Wei Gong. 2024. Bitalign: Bit Alignment for Bluetooth Backscatter Communication. *IEEE Transactions on Mobile Computing* (2024), 1–11. <https://doi.org/10.1109/TMC.2024.3374815>
- [14] Zhanxiang Huang and Wei Gong. 2022. EAScatter: Excitor-Aware Bluetooth Backscatter. In *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*. 1–10. <https://doi.org/10.1109/IWQoS54832.2022.9812894>
- [15] Infineon. 2011. Silicon Schottky Diode. https://www.infineon.com/dgdl/Infineon-BAT63SERIES-DS-v01_01-en.pdf?fileId=db3a304314dca38901151817843c0df4. Accessed: July 2024.
- [16] Vikram Iyer, Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua Smith. 2016. Inter-Technology Backscatter: Towards Internet Connectivity for Implanted Devices. In *Proceedings of the 2016 ACM SIGCOMM Conference (Florianopolis, Brazil) (SIGCOMM '16)*. Association for Computing Machinery, New York, NY, USA, 356–369. <https://doi.org/10.1145/2934872.2934894>
- [17] Maoran Jiang and Wei Gong. 2023. Dances with Blues: Harnessing Multi-Frequency Carriers for Commodity Bluetooth Backscatter. *Proc. ACM Netw.* 1, CoNEXT3, Article 11 (nov 2023), 20 pages. <https://doi.org/10.1145/3629133>
- [18] Maoran Jiang and Wei Gong. 2024. Bidirectional Bluetooth Backscatter With Edges. *IEEE Transactions on Mobile Computing* 23, 2 (2024), 1601–1612. <https://doi.org/10.1109/TMC.2023.3241202>
- [19] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft. 2006. XORs in the air: Practical Wireless Network Coding. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. 243–254.
- [20] Keysight. 2017. MXR608A Infiniium MXR-Series Oscilloscope. <https://www.keysight.com/us/en/product/MXR608A/infiniium-mxr-series-oscilloscope-6-ghz-8-channels.html>. Accessed: July 2024.
- [21] Songfan Li, Chong Zhang, Yihang Song, Hui Zheng, Lu Liu, Li Lu, and Mo Li. 2020. Internet-of-microchips: Direct radio-to-bus communication with SPI backscatter. In *Proceedings of the 26th annual international conference on mobile computing and networking*. 1–14.
- [22] Wenyuan Ma, Wei Wang, and Tao Jiang. 2020. Joint energy harvest and information transfer for energy beamforming in backscatter multiuser networks. *IEEE Transactions on Communications* 69, 2 (2020), 1317–1328.
- [23] Qianhe Meng, Han Wang, Chong Zhang, Yihang Song, Songfan Li, Li Lu, and Hongzi Zhu. 2024. Processor-Sharing Internet of Things Architecture for Large-scale Deployment. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*. 211–224.
- [24] Microsemi. 2019. AGLN060 Datasheet. <https://www.microsemi.com/product-directory/parts/142383>. Accessed: July 2023.
- [25] Xin Na, Xiuzhen Guo, Zihao Yu, Jia Zhang, Yuan He, and Yunhao Liu. 2023. Leggiero: Analog WiFi backscatter with payload transparency. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*. 436–449.
- [26] Natalya Naumenko. 2019. LiNbO₃ plate bonded to quartz as a substrate for high frequency wideband SAW devices. In *2019 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 1227–1230.
- [27] Ryu Okubo, Luke Jacobs, Jinhua Wang, Steven Bowers, and Elahe Soltanaghai. 2024. Integrated Two-way Radar Backscatter Communication and Sensing with Low-power IoT Tags. In *Proceedings of the ACM SIGCOMM 2024 Conference*. 327–339.
- [28] Bozheng Pang, Tim Claeys, Jens Vankeirsbilck, Kristof T'Jonck, Hans Hallez, and Jeroen Boydens. 2022. A Novel Model to Quantify the Impact of Transmission Parameters on the Coexistence Between Bluetooth Low Energy Pairs. *IEEE Internet of Things Journal* 10, 2 (2022), 1733–1745.
- [29] Sara M Reyes, Chloe Fong, James D Rosenthal, and Matthew S Reynolds. 2024. ZeroScatter BLE: All-Digital, Fully Synthesizable Bluetooth Low Energy (BLE) Backscatter Communication in Verilog RTL for FPGA-Based Devices. In *2024 IEEE International Conference on RFID (RFID)*. IEEE, 7–10.
- [30] Mohammad Rostami, Xingda Chen, Yuda Feng, Karthikeyan Sundaresan, and Deepak Ganesan. 2021. MIXIQ: Re-thinking ultra-low power receiver design for next-generation on-body applications. In *Proceedings of the 27th annual international conference on mobile computing and networking*. 364–377.
- [31] Mike Ryan. 2013. Bluetooth: With Low Energy Comes Low Security. In *7th USENIX Workshop on Offensive Technologies (WOOT '13)*. USENIX Association, Washington, D.C. <https://www.usenix.org/conference/woot13/workshop-program/presentation/ryan>

- [32] Nordic Semiconductor. 2023. nRF52833 DK Datasheet. <https://www.nordicsemi.com/Products/Development-hardware/nrf52833-dk/getstarted>. Accessed: July 2023.
- [33] Nordic Semiconductor. 2024. nRF52833 Product Specification. <https://docs.nordicsemi.com/bundle/psnrf52833/page/keyfeatureshtml5.html>. Accessed: Dec 2024.
- [34] NXP Semiconductors. 2022. Raising the Bar in Security and Performance for Automotive Bluetooth® Low Energy. <https://www.nxp.com/company/about-nxp/smarter-world-blog/BL-AUTOMOTIVE-BLE>. Accessed: July 2024.
- [35] Bluetooth SIG. 2016. Bluetooth Core Specification. <https://www.bluetooth.com/specifications/specs/core-specification-5-0>. Accessed: July 2023.
- [36] Bluetooth SIG. 2023. LE Audio: The Future of Bluetooth Audio. <https://www.bluetooth.com/bluetooth-resources/mrn-le-audio/>. Accessed: July 2023.
- [37] Bluetooth SIG. 2024. The Ambient IoT: The Emergence of a New Class of Bluetooth IoT Devices. https://www.bluetooth.com/wp-content/uploads/2024/01/MRN_Ambient-IOT.pdf. Accessed: March 2024.
- [38] Yihang Song, Li Lu, Jiliang Wang, Chong Zhang, Hui Zheng, Shen Yang, Jinsong Han, and Jian Li. 2023. μ Mote: Enabling Passive Chirp De-spreading and μ W-level Long-Range Downlink for Backscatter Devices. In *20th USENIX symposium on networked systems design and implementation (NSDI '23)*. 1751–1766.
- [39] STMicroelectronics. 2024. Comparator Datasheet. <https://www.semiee.com/file2/ea64b463a4c8fda58ae359216196b32d/ST/ST-TS3021.pdf>. Accessed: September 2024.
- [40] Kasyap Suresh, Varun Jeoti, Socheatra Soeung, Micheal Drieberg, Mark Goh, and Muhammad Zubair Aslam. 2020. A Comparative Survey on Silicon Based and Surface Acoustic Wave (SAW)-Based RFID Tags: Potentials, Challenges, and Future Directions. *IEEE Access* 8 (2020), 91624–91647.
- [41] Synopsis. 2017. Synopsis Design Compiler. <http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DesignCompiler/Pages/default.aspx>. Accessed: July 2023.
- [42] Thierry Taris, Jean-Baptiste Begueret, and Yann Deval. 2011. A 60μ W LNA for 2.4 GHz wireless sensors network applications. In *2011 IEEE Radio Frequency Integrated Circuits Symposium*. IEEE, 1–4.
- [43] Ambuj Varshney, Andreas Soleiman, and Thiemo Voigt. 2019. TunnelScatter: Low Power Communication for Sensor Tags using Tunnel Diodes. In *The 25th annual international conference on mobile computing and networking*. 1–17.
- [44] Maolin Zhang, Si Chen, Jia Zhao, and Wei Gong. 2021. Commodity-level BLE backscatter. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 402–414.
- [45] Maolin Zhang, Jia Zhao, Si Chen, and Wei Gong. 2020. Reliable Backscatter with Commodity BLE. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 1291–1299.
- [46] Pengyu Zhang, Colleen Josephson, Dinesh Bharadia, and Sachin Katti. 2017. FreeRider: Backscatter Communication Using Commodity Radios. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. 389–401.
- [47] Zhang Zhang, Tiejun Lv, Xin Su, and Hui Gao. 2011. Dual XOR in the Air: A Network Coding Based Retransmission Scheme for Wireless Broadcasting. In *2011 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.
- [48] Renjie Zhao, Fengyuan Zhu, Yuda Feng, Siyuan Peng, Xiaohua Tian, Hui Yu, and Xinbing Wang. 2019. OFDMA-enabled Wi-Fi backscatter. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–15.