Wook: Enabling High-Throughput Wi-Fi Downlink with Ultra-Low Power

Bingbing Wang, Zeming Yang, Wenhui Li, Linling Zhong, Fengyuan Zhu, Jiazhen Lei, Jianyu Luo, Meng Jin, and Xiaohua Tian[*]
Shanghai Jiao Tong University
{wangbingbing,yzm_sjtu,lwh1999,uranium_zll,jsqdzhufengyuan,leijz,LuoJY_0515,jinm,xtian}@sjtu.edu.cn

ABSTRACT

The Wi-Fi-enabled ultra-low power communication system exhibits high asymmetry between uplink and downlink speeds. The uplink can reach up to 1 Mbps, while the downlink throughput is around 100 Kbps. In this paper, we present *Wook*, a novel high throughput downlink system to empower Commercial Off-The-Shelf (COTS) **W**i-Fi devices to transmit high-speed **OOK** messages. The key innovation underpinning *Wook* is its ability to achieve sub-symbol level modulation, allowing a single OFDM symbol to carry multiple OOK bits. This is done by profoundly exploring the Wi-Fi PHY layer and identifying optimal input payload to achieve fine-grained Wi-Fi waveform manipulation. We fabricate a PCB prototype and employ the COTS Wi-Fi router to implement the entire system. Experimental results show that with a simulated IC power consumption 76.6μW, *Wook* achieves a data rate of up to 1.1 Mbps, an 8.9X improvement over state-of-the-art systems. Moreover, even at a communication distance of 95 m, *Wook* maintains a throughput of 82.9 Kbps.

CCS CONCEPTS

• **Networks** → **Network architectures**; • **Hardware** → **Wireless devices**.

KEYWORDS

Wi-Fi, downlink, ultra-low power.

[*]Corresponding author.

ACM MOBICOM '25, November 4–8, 2025, Hong Kong, China

ACM ISBN 979-8-4007-1129-9/2025/11…$15.00
https://doi.org/10.1145/3680207.3723475

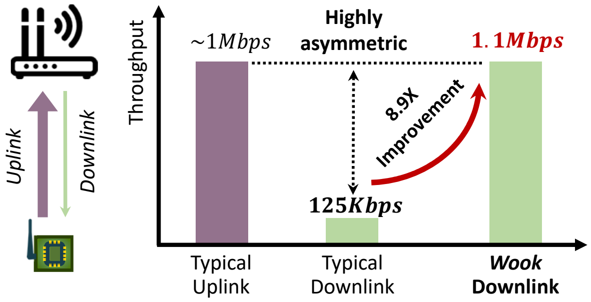


Figure 1: *Wook* address the throughput asymmetry issue in ULP Wi-Fi IoT network.

ACM Reference Format:
Bingbing Wang, Zeming Yang, Wenhui Li, Linling Zhong, Fengyuan Zhu, Jiazhen Lei, Jianyu Luo, Meng Jin, and Xiaohua Tian. 2025. *Wook*: Enabling High-Throughput Wi-Fi Downlink with Ultra-Low Power. 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/10.1145/3680207.3723475

1 INTRODUCTION

Ubiquitous Wi-Fi signals propel the Internet of Things (IoT) network to unprecedented levels of connectivity. The global Wi-Fi IoT market is expected to reach an estimated $3.2 billion by 2030 with a CAGR of 18.7% from 2024 to 2030 [1]. However, a Wi-Fi PHY-compatible system typically requires tens to hundreds of milliwatts [2], which becomes a bottleneck for the further development of Wi-Fi IoT.

As shown in Fig. 1, the ultra-low power (ULP) uplink systems, like Wi-Fi backscatter, claim to achieve Mbps-level data transmission with 10s μW by passively reflecting excitation signals [3–9]. However, as shown in Table 1, existing ULP downlink systems [10–17] struggle to achieve a throughput of ~100 Kbps. This disparity arises because uplink demodulation relies on high-end devices, such as COTS Wi-Fi devices or software-defined radios (SDRs), to process uplink signals and demodulate complex modulations like QPSK, 16-PSK, and DSSS. In contrast, ULP receivers typically use an envelope detector (ED) followed by a comparator, as shown in Fig. 2, which can only demodulate simple OOK signals. For an ultra-low-power Wi-Fi transceiver, a high-speed downlink is as crucial as a high-speed uplink for several reasons: 1)

Table 1: Comparing *Wook* with low-power downlink systems in IoT devices.

	System Name/Type	Power	Data rate	Range
Low Power	4G/5G	100s mW	10s Mbps	Long
	Wi-Fi	10s mW	10s Mbps	Medium
	LTE-M	10s mW	~1 Mbps	Long
	LoRa/NB-IoT	10s mW	10s Kbps	Long
	BLE/ZigBee	~10mW	100s Kbps	Short
Ultra-low Power	Packet-level [10–12]	10s μW	10s Kbps	Short
	Symbol-level [13]	10s μW	125 Kbps	Short
	Saiyan [14]	10s μW	10s Kbps	Medium
	Passive DSSS [15]	10s μW	~10 Kbps	Medium
	μmote [16]	10s μW	~10 Kbps	Medium
	Analog OFDMA [17]	10s μW	~100 Kbps	Short
	Wook	**10s μW**	**1.1 Mbps**	**Medium**

it reduces the operating time required to transmit the same amount of data, thereby decreasing energy consumption; 2) it lowers transmission latency; and 3) it enables transmitting richer media content, such as images and even videos.

In this paper, we try to answer the following question: *Is it possible to reach a Mbps downlink throughput with ultra-low power in Wi-Fi environments?* To seamlessly integrate with the backscatter systems, the high throughput design should meet two critical requirements: 1) the wake-up circuits in the backscatter tags can be utilized as the downlink receiver, and 2) compatible with COTS Wi-Fi devices. To meet the first requirement, the transmitters should transmit OOK messages. For the second requirement, a naive strategy involves controlling Wi-Fi devices whether or not to send Wi-Fi packets to represent '1' or '0', respectively [11, 18]. This packet-level modulation exhibits low efficiency, resulting in only 10s Kbps throughput. An alternative approach is symbol-level modulation [13], where a bit is represented by altering the order of two OFDM symbols with differing energies. This allows for a throughput of up to 125 Kbps but still far below the uplink throughput.

We present *Wook*, a novel system allows IoT devices to demodulate high-speed Wi-Fi signals with ultra-low power. Specifically, *Wook* achieves high throughput by utilizing sub-symbol level modulation, i.e., embedding multiple OOK bits into a single OFDM symbol. As shown in Fig. 1, *Wook*'s throughput goes up to 1.1 Mbps, eliminating the throughput asymmetry between the uplink and downlink. The key lies *Wook* generates the payload to manipulate the Wi-Fi waveform in a fine-grained manner. As shown in Fig. 3, this manipulation allows the Wi-Fi waveform to exhibit multiple stable energy shifts within an OFDM symbol. To realize *Wook*, we need to address three key challenges.

Challenge 1: How to shape the Wi-Fi waveform? Demodulating Wi-Fi signals is power-hungry. The receiver initially detects the STF for frame synchronization and then uses the LTF for channel equalization. After correcting phase errors with pilots, the QAM symbols are demapped, and the payload is recovered after channel decoding. COTS Wi-Fi chips consume 10s mW. In comparison, *Wook* terminals with simple hardware (see Fig. 2) consuming 10s μW can only differentiate between strong and weak signal energies in a binary manner, rendering them incapable of demodulating complex OFDM signals. To enable sub-symbol level modulation, we aim for the Wi-Fi waveform to exhibit continuous and stable changes within an OFDM symbol. Firstly, the waveform should display a significant energy difference between the high-energy and low-energy regions to facilitate adaptive threshold adjustment. Secondly, the high-energy regions should maintain a higher energy level to reduce energy loss. Thirdly, the energy of each sample in a region should be approximately identical to avoid decision errors. We note that each sample in the Wi-Fi waveform is obtained by weighted superposition of the data on all subcarriers. For example, an 802.11n 20MHz channel includes 52 data subcarriers. When 64-QAM modulation is applied, there can be 64^{52} possible waveforms, making exhaustive enumeration infeasible. To address this, *Wook* iteratively optimizes the Wi-Fi waveform by formulating a versatile goal function and adopting hybrid heuristic algorithms to determine the optimal QAM symbols for each data subcarrier.

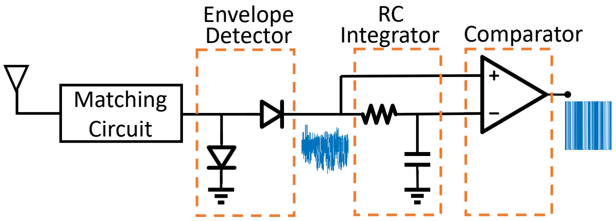


Figure 2: Ultra-low power OOK demodulator.

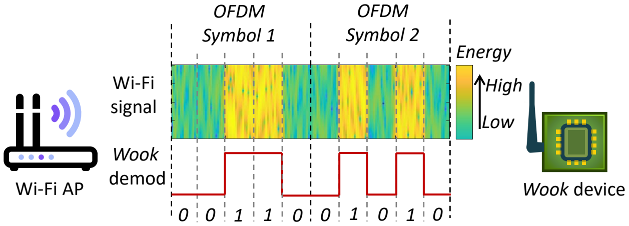
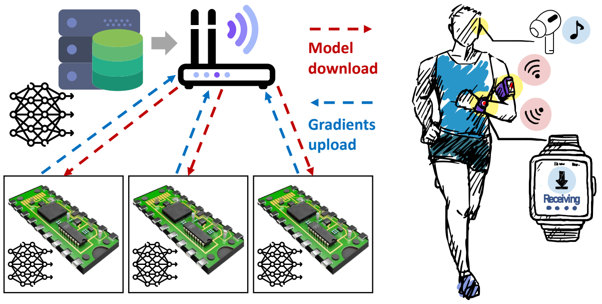


Figure 3: *Wook* enables sub-symbol level modulation for higher throughput.

Challenge 2: How to generate the Wi-Fi payload? Unlike SDRs, COTS Wi-Fi devices use specialized SoC to run the 802.11 protocol, which restricts users from directly writing IQ data for transmission. As a result, we can only control the Wi-Fi waveform indirectly by controlling the input payload of the Wi-Fi transmission flow. Although the overall Wi-Fi waveform is a composite of the waveforms of individual OFDM symbols, simply cascading the bits (i.e., the demodulated bits of the QAM symbols derived in Challenge



(a) Federated learning. (b) Wearable devices.

Figure 4: *Wook* application cases.

1) of each OFDM symbol as the input payload does not yield the desired waveform. The reason is that before the QAM mapping, the input bits undergo padding, scrambling, channel coding, and bit interleaving, as shown in Fig. 5, these processes completely restructure the input bits. The conventional methods [13, 19, 20] run the 802.11 reverse engineering to derive the payload. However, they treat the entire OFDM symbol as a single unit and can only achieve coarse symbol-level modulation. We reveal that the bottleneck on the ULP downlink lies in the overlooked channel coding. Specifically, the previous systems use binary convolutional code (BCC), which is memorized and causes the output waveform not to be manipulated at a fine-grained level. We note that 802.11 offers an optional coding scheme, i.e., low-density parity-check (LDPC) code, which is memory-less and capable of preserving the integrity of the input information bits. *Wook* innovatively conducts 802.11 forward analysis to derive a Wi-Fi payload that enables the COTS Wi-Fi devices to send high-speed OOK messages.

Challenge 3: How to demodulate high-order OOK symbols? OOK is highly susceptible to noise interference as a type of amplitude modulation. The conventional method, which directly uses the output of the comparator as the demodulation result, often produces a high bit error rate (BER). Complex modulations like 16-QAM use correlation demodulation to reduce the BER [21]. Our key insight is to treat all OOK bits represented by an OFDM symbol as a high-order OOK symbol. For example, a 32-OOK symbol signifies there are $log_2 32 = 5$ OOK bits contained in one OFDM symbol. Based on this, we propose a correlation demodulation method tailored to *Wook*'s sub-symbol level modulation. This method effectively enhances the system's noise immunity.

To summarize, we make the following contributions through the *Wook* design and implementation.

- We introduce *Wook*, a novel design enabling COTS Wi-Fi devices to transmit high-speed OOK messages. *Wook* addresses the asymmetry in communication rates between uplink and downlink in ULP Wi-Fi IoT networks, promoting the broader adoption of battery-free IoT devices.
- We propose a high throughput Wi-Fi downlink paradigm by deeply exploring the 802.11 PHY to iteratively produce the desired waveforms with heuristic algorithms. Crucially, *Wook* integrates seamlessly with existing ULP devices, thus avoiding additional hardware overhead.
- We prototype *Wook* with COTS components and conduct comprehensive experiments with a COTS Wi-Fi router in real-world scenarios. The results demonstrate that *Wook* achieves a data rate of up to 1.1 Mbps, surpassing state-of-the-art solutions by a factor of 8.9.

2 APPLICATION CASES

Table 1 illustrates that conventional low-power systems consume tens to hundreds of milliwatts, requiring frequent battery replacements or recharging, which constrains the scalability of IoT networks. *Wook* facilitates high-speed downlink data transmission with ultra-low power, making it suitable for diverse applications.

Tiny machine learning (TinyML) devices. TinyML integrates deep learning models into low-power microcontrollers, enabling ubiquitous intelligence [22]. These devices are engineered for low power consumption and are generally powered by batteries. Downlink is essential for updating model parameters [23, 24] or performing over-the-air (OTA) firmware updates [25], but the associated communication power overhead has emerged as a significant bottleneck in TinyML development [24, 26, 27]. Widely adopted ZigBee, LoRa, and NB-IoT often exhibit power consumption levels that are equivalent to or surpass those of microcontrollers, hence diminishing the energy allocated for computation and increasing data processing latency. Moreover, frequent battery recharging elevates deployment costs. *Wook* overcomes these challenges by offering high energy efficiency and conserving more energy for the microcontrollers. For example, federated learning (FL) [28] collaborates with TinyML by enabling on-device training [23, 24]. As shown in Fig. 4(a), FL involves frequent data exchanges: 1) TinyML devices either download an initialized model from a central server or collaboratively construct one, and 2) during the training process, TinyML devices frequently download the latest model parameters from the central server. *Wook* can accelerate the training process in three aspects: 1) its high throughput minimizes communication time overhead, 2) its low power consumption allows the microcontroller to operate at peak clock frequency for faster computation, and 3) it prolongs battery life and reduces additional control overhead.

Wearable devices. Battery-operated wearable devices commonly use BLE for continual connection with smartphones. For examples, as shown in Fig. 4(b), wireless earphones require high throughput downlink to receive high-quality audio signals, and smart watches need to synchronize notifications with smartphones or download the large

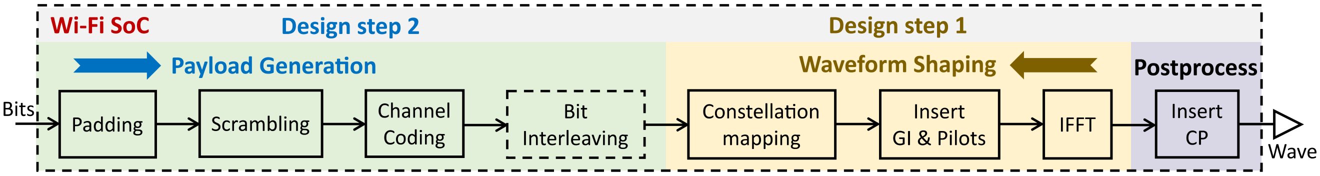


Figure 5: 802.11 transmission flow.

OTA firmware. As shown in Table 1, BLE consumes around 10mW in receive-only mode, whereas *Wook* consumes only 1% of the power BLE requires while maintaining comparable throughput. Thus, adopting *Wook* for receiving downlink data helps extend the battery life of wearable devices.

3 PRELIMINARY

Before delving deeper into *Wook*, it is essential to know the ULP OOK demodulator and the 802.11 transmission flow.

3.1 OOK Demodulator

OOK demodulators are widely used in backscatter tags to receive wake-up signals. As shown in Fig. 2, a OOK demodulator includes an ED followed by a voltage comparator. When RF signals reach the antenna, the ED converts the RF signals into low-frequency signals with varying amplitudes. The comparator then generates binary 0s and 1s depending on whether the amplitude is below or exceeds the threshold. To dynamically change the threshold, a simple RC integrator is employed [15, 16]. Such a hardware architecture offers high energy efficiency and ease of assembly [29]. Contemporary designs increasingly adopt application-specific integrated circuits to improve sensitivity, which we will show in Sec. 8.

3.2 802.11 Transmission Flow

The 802.11 transmission flow is outlined in Fig. 5.

Padding. The input bit length is random, but the number of bits in a codeword is fixed. Wi-Fi pad each codeword with 0s to meet the bit length specifications.

Scrambling. Scrambling entails the XOR operation between the input bit sequence and a scrambling sequence. The scrambled output are entirely distinct from the input.

Channel coding. The 802.11 standards offer two channel coding schemes, i.e., BCC and LDPC. We note that 802.11 uses system LDPC codes, where information bits and check bits are distinctly segregated within a codeword [30]. In contrast, BCC fully reorganizes the information bits [21]. The number of OFDM symbols in a Wi-Fi frame is an integer, while the number of bits in each OFDM symbol remains constant for a specific PHY configuration. Wi-Fi performs puncturing or repeating operations to address this alignment issue.

Bit interleaving. Bit interleaving rearranges the input bits to mitigate burst errors, which can severely degrade BCC's error-correcting performance but have minimal effect on LDPC. Notably, Wi-Fi disables bit interleaving when LDPC is the coding scheme.

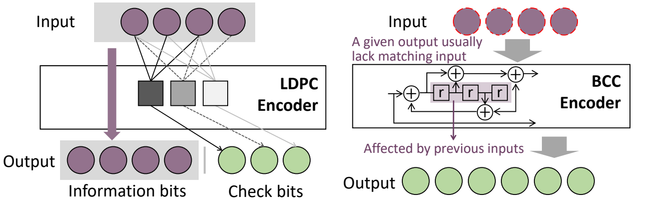


Figure 6: 1) LDPC can directly produce the desired output, while BCC cannot reliably find the correct input for the desired output. 2) The output of BCC is related to the history input.

Constellation mapping. This process involves mapping bits to QAM symbols. Each data subcarrier within an OFDM symbol contains a QAM symbol.

Guard intervals and pilots. Besides data subcarriers, OFDM symbols include guard intervals (GIs) and pilots. For example, in 802.11n/ac, a 20 MHz channel contains 52 data subcarriers, 4 pilots, and 8 GIs. The GIs are populated with 0s. The pilots are assigned with 1s or -1s, which is fixed for a specific PHY configuration.

IFFT. The IFFT converts the data on each subcarrier into time domain samples:

$$t_n = \frac{1}{N^{sc}} \sum_{k=1}^{N^{sc}} d_k e^{-j\frac{2\pi}{N^{sc}}(n-1)k}, \quad n = 1, \dots N^{sc}, \tag{1}$$

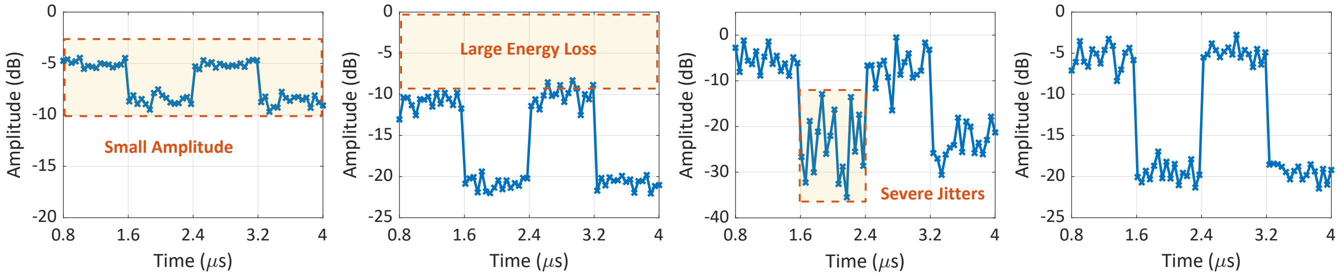
where N^{sc} denotes the number of subcarriers, t_n denotes the n^{th} sample, and d_k denotes the data on the k^{th} subcarrier. Each sample can be viewed as the weighted cumulative sum of all subcarrier data. Therefore, *any changes made to the data in a subcarrier will affect entire the OFDM symbol's waveform.*

Cyclic prefix. To mitigate inter-symbol interference (ISI), Wi-Fi prepends a segment of the last 1/4 or 1/8 samples to the front. The CP does not convey useful information and is ignored by the receiver.

4 DESIGN METHODOLOGY

Wook finds the input information bits to control the waveform of each OFDM symbol, enabling sub-symbol level modulation. As shown in Fig. 6, using LDPC instead of BCC is crucial for this process.

- BCC completely changes the information bits, retrieving the corresponding information bits from the desired output is necessary. However, since this desired output is unlikely to form a valid codeword, the decoding process is prone



(a) Small amplitude

(b) Large energy loss

(c) Severe jitters

(d) Optimization results

Figure 7: Goal function design issues to implement high order OOK modulation.

to failure. In contrast, LDPC preserve the information bits, eliminating the need for a reverse process.

- Unlike BCC, where the output depends on previous inputs, LDPC is memory-less, meaning the encoded bits are determined solely by the information bits within each block. This feature enables independent control of the QAM symbols on each OFDM symbol.
- The effects of bit interleaving do not need to be considered.

Previous work routinely uses BCC, which provides only coarse control over the Wi-Fi waveform and barely achieves symbol-level modulation.

The LDPC codes in 802.11 have a maximum codeword length of 1944. In a 20 MHz channel, the 52 data subcarriers within a single OFDM symbol can carry $52 \times log_2 64 = 312$ bits. In conjunction with the information present in Sec. 3.2, we derive several key insights: 1) the Wi-Fi waveform comprises concatenated waveforms from each OFDM symbol; 2) the waveform of an OFDM symbol is governed by the QAM symbols onto its data subcarriers; 3) the system LDPC codes can preserve the original information bits in the codewords; and 4) the information bit length of an LDPC codeword longer than the bit length carried in an OFDM symbol.

Building on the above insights, the design of *Wook* is formulated in three steps. Firstly, leveraging insights 1) and 2), we determine the optimal QAM symbols within an OFDM symbol to achieve sub-symbol level modulation. Secondly, leveraging insights 3) and 4), we generate the input payload to form a valid Wi-Fi frame. The design methodology for these two steps is depicted in Fig. 5. The final step is to recover the OOK messages with ULP receivers.

5 DESIGN

This section shows the design details of *Wook*. We here denote M^{ook}-OOK modulation as a single OFDM symbol conveys $K^{ook} = log_2 M^{ook}$ OOK bits.

5.1 Wi-Fi Waveform Shaping

The Wi-Fi waveform is shaped by the OFDM symbols, which are defined by the QAM symbols on their data subcarriers. We formulate the waveform shaping as an optimization problem. Since the CP does not convey valid information, we omit it in the following analysis.

5.1.1 Problem Formulation. The time-domain samples of an OFDM symbol $\mathbf{T} = \{t_1, t_2, ..., t_{N^{sc}}\}$ are calculated by Equ. 1. To convey K^{ook} bits in one OFDM symbol, we partition $\mathbf{T}$ into K^{ook} groups, with each group containing $n^{ook} = \lfloor \frac{N^{sc}}{K^{ook}} \rfloor$ samples. The desired waveform can be defended as:

$$\mathbf{T}^d = \{\underbrace{b_1, \ldots, b_1}_{n^{\mathrm{ook}}}, \ldots, \underbrace{b_{K^{\mathrm{ook}}}, \ldots, b_{K^{\mathrm{ook}}}}_{n^{\mathrm{ook}}}, \ldots, t_{N^{\mathrm{sc}}}\}, \tag{2}$$

where $b_i \in \{0, 1\}$ denotes the i^{th} OOK bit. Then, we formulate the optimization as follows:

$$\min_{d'_1,...,d'_{N^{ds}}} \sum_{n=1}^{K^{ook} n^{ook}} \left| |t_n| - t_n^d \right|. \tag{3}$$

Here, N^{ds} denotes the number of data subcarriers, t_n^d denotes the n^{th} element of $\mathbf{T}^d$. The complex data on the i^{th} data subcarrier, denoted as d'_i, which satisfies:

$$\{d'_1, ..., d'_{N^{ds}}\} \subseteq \{d_1, ..., d_{N^{sc}}\}, \tag{4}$$

Each d'_i is determined by:

$$d'_i = \mathbf{Q}\left(m_i\right) \qquad m_i \in \{1, 2, \ldots, M^{qam}\}, \tag{5}$$

where $\mathbf{Q}$ is a complex constant vector consisting of M^{qam} elements, and M^{qam} is the QAM modulation order. Each element in $\mathbf{Q}$ represents a complex constellation point of an M^{qam}-QAM symbol. $\mathbf{Q}\left(m\right)$ denotes the m^{th} element of the $\mathbf{Q}$. Subsequently, Equ. 3 can be redefined as:

$$\min_{m_1,...,m_{N^{ds}}} \sum_{n=1}^{K^{ook} n^{ook}} \left| |t_n| - t_n^d \right|, \tag{6}$$

Optimizing Equ. 6 falls under the category of integer programming (IP) problem [20, 31], and there is no simple formula to calculate the optimal results. An exact solution can be obtained through exhaustive search or branch-and-bound methods. However, with a configuration of 20 MHz channel and 64-QAM, the complexity of the exhaustive search is 64^{52}, which is intractable on almost all computing platforms.

5.1.2 Problem Solving. *Wook* employs heuristic algorithms to address the IP problem. However, relying solely on Equ. 6 as the goal function can hardly produce optimal results. For example, as shown in Fig. 7, several challenges are present when encoding the 16-OOK symbol '1010'.

- Small amplitude. As shown in Fig. 7(a), heuristic algorithms tend to converge to local optima, resulting in a slight difference in amplitude between regions representing 1s and 0s. Such minor amplitude can obscure the decision threshold, increasing the likelihood of decision errors.
- Large energy loss. As shown in Fig. 7(b), heuristic algorithms may increase amplitude by compromising the overall energy. Excessive energy loss can substantially reduce the communication range.
- Amplitude jitters. As shown in Fig. 7(c), samples representing identical OOK bits can exhibit significant amplitude jitters. This instability in the waveform heightens susceptibility to noise interference, thereby increasing the BER.

We separate two sub-vectors $\mathbf{T}^0$ and $\mathbf{T}^1$ from the vector $\mathbf{T}$, representing samples for 0s and 1s, respectively. Let n^0 and n^1 denote the number of elements in $\mathbf{T}^0$ and $\mathbf{T}^1$, accordingly, such that $n^0 + n^1 = K^{ook}n^{ook}$. To address these challenges, *Wook* employs the following strategies.

Optimization goal 1: maximum amplitude difference. The goal is formulated through the following goal function:

$$GF_1 = P^{de} + \frac{1}{n^0}\sum_{i=1}^{n^0} 20lg\left|t_i^0\right| - \frac{1}{n^1}\sum_{j=1}^{n^1} 20lg\left|t_j^1\right|, \quad (7)$$

where P^{de} represents the targeted energy difference in decibels, and t_i^0 and t_j^1 denote the i^{th} element of $\mathbf{T}^0$ and the j^{th} element of $\mathbf{T}^1$, respectively.

Optimization goal 2: minimize energy loss. The primary goal is to maximize the normalized energy of samples representing 1s.

$$GF_2 = \sum\nolimits_{j=1}^{n^1}\left(1 - \left|t_j^1\right|\right), \quad (8)$$

Optimization goal 3: weaken energy jitters. The energy of samples representing the same bits should have minimal standard deviation.

$$GF_3 = \beta_0 \operatorname{std}\left(20lg\left|\mathbf{T}^0\right|\right) + \beta_1 \operatorname{std}\left(20lg\left|\mathbf{T}^1\right|\right), \quad (9)$$

where std denotes the operation of taking the standard deviation of the vector. The parameters β_0 and β_1 enable *Wook* to control the energy jitters in a fine-grained manner.

We then reformulate the global goal function as follows:

$$GF_{normal} = \min_{m_1,\ldots,m_{Nds}} \alpha_1 GF_1 + \alpha_2 GF_2 + \alpha_3 GF_3, \quad (10)$$

where α_1, α_2 and α_3 are penalty factors. By adopting Equ. 10 as the goal function in place of Equ. 6, we can derive the desired waveform, as demonstrated in Fig. 7(d).

However, when the OOK symbols consist of all 1s or 0s, some of the parameters in Equ. 10 are disabled. We analyze these two special issues below.

All 1s OOK symbol. As this configuration comprises only the bit '1', calculating the energy difference becomes irrelevant, i.e., GF_1 is unnecessary. While GF_2 remains unchanged, and GF_3 is adjusted to:

$$GF_3^{1s} = \operatorname{std}\left(20lg\left|\mathbf{T}^1\right|\right). \quad (11)$$

Accordingly, the goal function is defined as:

$$GF_{1s} = \min_{m_1,\ldots,m_{Nds}} \alpha_2 GF_2 + \alpha_3 GF_3^{1s}. \quad (12)$$

All 0s OOK symbol. In this scenario, GF_1 is also redundant. Adjustments to GF_2 and GF_3 are necessary.

$$GF_2^{0s} = \sum\nolimits_{i=1}^{n^0}\left|t_i^0\right|. \quad (13)$$

$$GF_3^{0s} = \operatorname{std}\left(20lg\left|\mathbf{T}^0\right|\right). \quad (14)$$

The goal function thereby updates to:

$$GF_{0s} = \min_{m_1,\ldots,m_{Nds}} \alpha_2 GF_2^{0s} + \alpha_3 GF_3^{0s}. \quad (15)$$

To convey K^{ook} OOK bits using one OFDM symbol, we only need to find M^{ook} OFDM symbols whose waveforms align with M^{ook} distinct OOK symbols. These OFDM symbols are then concatenated to assemble the complete Wi-Fi frame.

5.1.3 Optimization Flow. Simulated annealing (SA) is a metaheuristic algorithm designed to approximate the global optimum in an ample search space for complex optimization problems. While SA can theoretically find the global optimum in landscapes with many local optima, its effectiveness heavily depends on the initial position, influencing both convergence speed and the ability to avoid local minima [32]. The genetic algorithm (GA) is renowned for its proficiency in exploring extensive spaces through diverse potential solutions [33]. This insight leads us to integrate GA with SA, i.e., GA first explores the search space to identify promising regions, and then SA refines the search within these areas to pinpoint the optimal solution. This hybrid approach combines the strengths of both algorithms, enhancing efficiency and effectiveness in finding an optimal solution.

The hybrid algorithm may need to run 1 to 3 times with varying parameters to obtain the final results. In our experience, α_1 and α_3 typically range from 1 to 2, with typical values of 1.85 and 1.2, respectively, while α_2 ranges from 1 to 4, with a typical value of 2. The values of β_1 and β_2 correspond to the number of 0s and 1s in the target OOK symbol, respectively. The target energy difference p^{de} is 35 dB.

It is worth noting that the hybrid algorithm operates offline. Upon determination of the M^{ook} QAM symbol groups, they are then stored in memory. To transmit a message to a

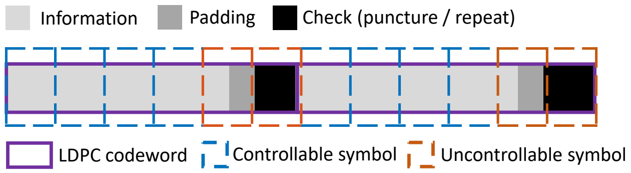


Figure 8: OFDM with LDPC coding.

ULP device, simply use a lookup table to select the appropriate QAM groups.

5.2 Wi-Fi Payload Generation

As shown in Fig. 8, the coding process involves not only information bits but also padding bits, check bits, and repeating or puncturing bits. However, only the information bits can be directly controlled by the payload. The principle of payload generation is to initialize a bit sequence and populate the regions where the controllable OFDM symbols are located with the bits corresponding to the QAM symbols obtained in Sec. 5.1.2.

5.2.1 Finding the Controllable OFDM Symbols. For the sake of clarity, the relevant parameters are presented in Table 2.

Preliminary. The basic relationship between some of the parameters is presented below.

- $k^{info} = n^{cw} \times r$
- $N^{sym} = N^{ds} \times log_2 M^{qam}$
- $N^{cw} = \left\lceil \frac{N^{pre}}{k^{info}} \right\rceil$, where $\lceil * \rceil$ denotes upward rounding. This equation indicates that the number of codewords must be an integer and capable of encoding N^{pre} bits.
- $N^{ofdm} = \left\lceil \frac{N^{pre}}{r \times N^{sym}} \right\rceil$, which means the number of OFDM symbols must be integer and able to carry N^{pre} bits.
- $N^{frame} = N^{sym} \times N^{ofdm}$

Composition of N^{pre}. N^{pre} consists of four parts. 1) A 16-bit service field, wherein the initial 7 bits are allocated for storing the initial value of the scrambler. 2) MAC frame header. 3) The input bit sequence. 4) 32 CRC bits. Note that we can only control the input bit sequence.

Padding. To construct the N^{cw} codewords, the length of the bits to-be-encoded is $k^{info} \times N^{cw}$. Therefore, the total number of padding bits is $N^{pad} = k^{info} \times N^{cw} - N^{pre}$. The number of padding bits required for the i^{th} codeword is:

$$\begin{cases} N_i^{pad} = \left\lfloor \frac{N^{pad}}{N^{cw}} \right\rfloor + 1 & i = 1, \dots, rem\left(\frac{N^{pad}}{N^{cw}}\right) \\ N_i^{pad} = \left\lfloor \frac{N^{pad}}{N^{cw}} \right\rfloor & i = rem\left(\frac{N^{pad}}{N^{cw}}\right) + 1, \dots, N^{cw}, \end{cases} \tag{16}$$

where $\lfloor * \rfloor$ and *rem* denote the floor rounding and remainder operations, respectively. The padded bits are then fed into the LDPC encoder.

Puncturing and repeating. The padding bits are removed after LDPC encoding, thus $N^{post} = n^{cw} \times N^{cw} - N^{pad}$. Three cases will arise at this time.

Table 2: Parameters of 802.11 frame.

Parameter	Explanation
n^{cw}	LDPC codeword length
r	Coding rate
k^{info}	Information length in a codeword
N^{pre}	Number of bits to be encoded
N^{post}	Number of bits after encoding
N^{frame}	Number of bits in a frame
N^{ofdm}	Number of OFDM symbols in a frame
N^{pad}	Number of padding bits in a frame
N^{pun}	Number of punctured bits in a frame
N^{rep}	Number of repeated bits in a frame
N^{sym}	Number of bits in a OFDM symbol
N^{cw}	Number of LDPC codewords in a frame

Case 1: $N^{frame} > N^{post}$. In this case, there are insufficient encoded bits to form a Wi-Fi frame. To address this, Wi-Fi copy some of the preceding information bits and append them to the end of the check bits. This operation is known as repeating. The total number of repeating bits is $N^{rep} = N^{frame} - N^{post}$. Similar to Equ. 16, the number of repeating bits in i^{th} codeword is:

$$\begin{cases} N_i^{rep} = \left\lfloor \frac{N^{rep}}{N^{cw}} \right\rfloor + 1 & i = 1, \dots, rem\left(\frac{N^{rep}}{N^{cw}}\right) \\ N_i^{rep} = \left\lfloor \frac{N^{rep}}{N^{cw}} \right\rfloor & i = rem\left(\frac{N^{rep}}{N^{cw}}\right) + 1, \dots, N^{cw}. \end{cases} \tag{17}$$

Case 2: $N^{frame} < N^{post}$. Conversely, if the number of encoded bits exceeds the capacity of the Wi-Fi frame, Wi-Fi will remove some of the check bits to conform to the frame length. This process is known as puncturing. The total number of bits to be punctured is $N^{pun} = N^{post} - N^{frame}$. The number of punctured bits in i^{th} codeword is:

$$\begin{cases} N_i^{pun} = \left\lfloor \frac{N^{pun}}{N^{cw}} \right\rfloor + 1 & i = 1, \dots, rem\left(\frac{N^{pun}}{N^{cw}}\right) \\ N_i^{pun} = \left\lfloor \frac{N^{pun}}{N^{cw}} \right\rfloor & i = rem\left(\frac{N^{pun}}{N^{cw}}\right) + 1, \dots, N^{cw}. \end{cases} \tag{18}$$

Case 3: $N^{frame} = N^{post}$. The number of encoded bits exactly matches the Wi-Fi frame length requirements. No adjustments are necessary, i.e., $N^{rep} = N^{pun} = 0$.

After the padding, puncturing, and repeating operations, the number of information bits and check bits in the i^{th} codeword is determined as follows.

$$\begin{cases} k_i = k^{info} - N_i^{pad} \\ m_i = n^{cw}(1 - r) - N_i^{rep} + N_i^{pun}. \end{cases} \tag{19}$$

Note that only one of N_i^{rep} and N_i^{pun} can be non-zero.

The N^{frame} bits in a Wi-Fi frame can be segmented as:

$$N^{frame} = \{k_1, m_1, k_2, m_2, ..., k_{N^{cw}}, m_{N^{cw}}\}, \tag{20}$$

where $\sum_{i=1}^{N^{cw}} k_i = N^{pre}$. The N^{frame} bits can be equally divided into N^{ofdm} blocks, each corresponding to an OFDM

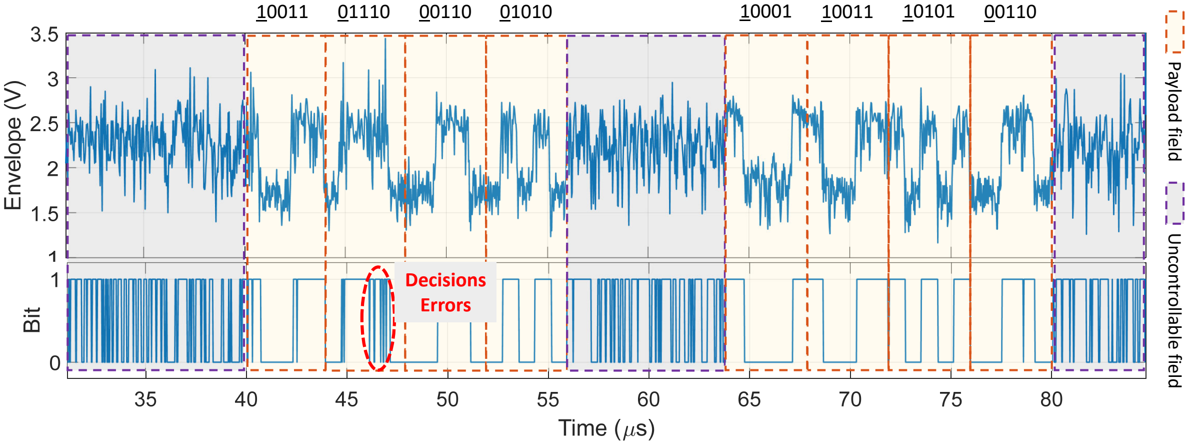
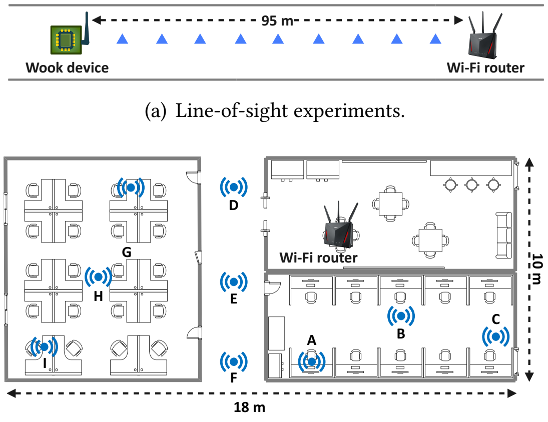


Figure 9: Envelope and its corresponding decision output.



(a) Line-of-sight experiments.

(b) Non-line-of-sight experiments.

Figure 10: Experiment scenarios.

symbol. The criteria to determine whether the i^{th} OFDM symbol is controllable are given by:

$$\begin{cases} N_i^{ofdm} & \text{Controllableonly contain information bits.} \\ N_i^{ofdm} & \text{Uncontrollable} \quad \text{otherwise.} \end{cases} \tag{21}$$

where $i \in \{2, ..., N^{cw}-1\}$ because the first ODFM symbol contains the service field and MAC header, and the last OFDM symbol contains CRC bits.

We let one OFDM symbol carry four OOK bits, each lasting $0.8\mu s$. When the *Wook* device is positioned 10 m away from the Wi-Fi AP, the output of the envelope detector and comparator is depicted in Fig. 9. The energy variations within the controllable field of the Wi-Fi signals are discernible and can be correctly recovered to OOK bits by the comparator. In contrast, the uncontrollable fields, which include the MAC header or check bits, exhibit such severe and irregular energy jitter that they cannot be used to transmit OOK messages.

5.2.2 Wi-Fi Payload Generation. After running the hybrid heuristic algorithms described in Sec. 5.1, we generate M^{ook} QAM symbol groups, with each group consisting of N^{ds} QAM symbols to form M^{ook} OFDM symbols, correspondingly represent M^{ook} OOK symbols. After demodulating these QAM symbol groups by obeying the 802.11 constellation mapping rule, we obtain M^{ook} bit groups, each containing N^{sym} bits. Next, the information bits in the controllable OFDM symbols - defined by Equ. 21 - are selected according to the downlink payload. For information bits in uncontrollable OFDM symbols, we set them randomly. This yields a payload bit sequence of length N^{pre}, denoted as $\mathbf{B}^{pre}$. However, directly inputting $\mathbf{B}^{pre}$ into the 802.11 transmission flow is ineffective owing to the scrambling operation.

Wi-Fi employs a 7-bit linear feedback shift register with polynomial x^7+x^4+1 to generate scrambled bits. The output of the scrambler is given by $\mathbf{B}^{post} = \mathbf{B}^{pre} \oplus \mathbf{S}$, where $\mathbf{S}$ represents the scrambling bit sequence. And $\mathbf{S}$ relies exclusively on the scrambler's initial value. Thus, by providing $\mathbf{B}^{post}$ as the input, we can derive the desired waveform in the output of the 802.11 transmission flow.

5.3 Correlation Demodulator

The conventional OOK demodulation relies on hard decision, where the comparator's output is directly taken as the demodulation result. However, this approach is more vulnerable to interference, especially in the congested 2.4 GHz band. For example, in Fig. 9, samples representing bit '1' may occasionally be misjudged as bit '0'.

In Wi-Fi backscatter, the tags use a 20 MHz clock to control a RF switch for frequency shifting [6, 34]. Therefore, without introducing extra hardware overhead, we can sample the comparator's output at a frequency of 20 MHz. Consequently, *Wook* employs a correlation decision to demodulate the OOK symbols. Specifically, *Wook* treats all bits conveyed by an OFDM symbol as a single OOK symbol. The correlation demodulator samples the output of the comparator to obtain all samples of an OOK symbol, which are then correlated with templates representing different OOK symbols. This

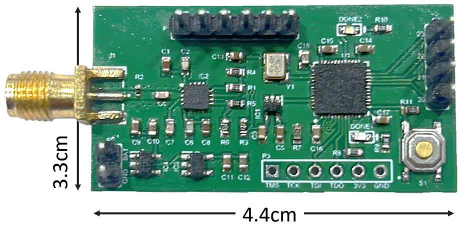


Figure 11: *Wook* PCB prototype.

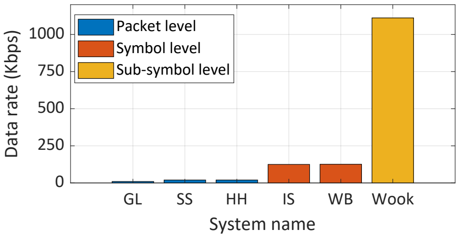


Figure 12: Data rate comparisons.

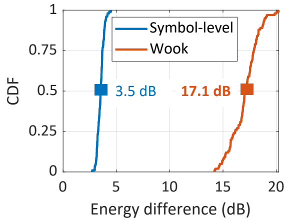


Figure 13: Energy differences comparison.

process is shown in Equ. 22.

$$s^{ook} = \underset{m\in\{1,2,\ldots,M^{ook}\}}{\arg\max} \sum_{n=1}^{K^{ook}n^{ook}} \mathrm{XNOR}\left(c_n, TEMP_{m,n}\right), \quad (22)$$

where c_n is n^{th} output of the comparator for an OOK symbol, and $TEMP$ is a predefined template matrix whose each row represents an OOK symbol. The demodulator selects the OOK symbol with the highest correlation coefficient as the demodulation result. We evaluate the performance gap between correlation decision and hard decision in Sec. 7.3.

6 IMPLEMENTATION

In this section, we show the implementation details of *Wook*.

6.1 Samples Utilization

To maximize resource efficiency, we aim to use all samples of controllable OFDM symbols to convey OOK bits. However, this is often not the case.

Cyclic prefix (CP). Wi-Fi duplicate the last 1/4 (normal) or 1/8 (short) valuable samples to the front of an OFDM symbol. As shown in Fig. 9, the bit carried by the first 1/5 part is identical to the last 1/5 part in an OFDM symbol. Since the CP does not carry new information, the receiver needs to disregard it. During evaluation, we set the Wi-Fi AP to short mode to improve the communication efficiency.

Idle samples. We notice that heuristic algorithms converge more readily if some samples are intentionally left unused, i.e., $n^{ook}K^{ook} < N^{sc}$ in Equ. 2. This slack constraint broadens the operational landscape of the algorithms. We utilize two configurations during our evaluations. 1) For 16-OOK modulation, each bit lasting $0.7\mu s$, leaving the remaining $0.4\mu s$ idle. 2) For 32-OOK modulation, each bit lasting $0.55\mu s$, leaving the remaining $0.45\mu s$ idle.

6.2 Synchronize with Wi-Fi AP

Wi-Fi synchronizes with receivers using short training field (STF) in the preamble. However, detecting STF is power-hungry. Instead, *Wook* utilizes the first 16 bits as a synchronization frame header, i.e., '0011_1110_0110_1010'. This 16-bit synchronization sequence includes a 13-bit barker code to reduce the mis-synchronization rate.

6.3 OOK Coding Rate

We define the ratio of controllable OFDM symbols to the total number of OFDM symbols in a Wi-Fi frame as the OOK coding rate, which depends on the number of data subcarriers N^{ds}, QAM modulation order M^{qam} and payload length N^{pre}. To balance exploring space and communication distance, we use 20 MHz channels and 64-QAM. The maximum data frame size for 802.11n is 2304 bytes, which means the maximum value of N^{pre} is 2304×8. To maximize downlink throughput while adhering to Wi-Fi's frame length constraints, we set $N^{pre} = 2115 \times 8$, achieving an OOK coding rate of 0.8 (66 OFDM symbols, with 53 of them being controllable).

6.4 Heuristic Algorithms

We run the hybrid heuristic algorithms in Sec. 5.1.3 on a Intel i9-12900H processor.

GA. The GA in MATLAB supports integer optimization. We establish a population size of 1300 and a maximum of 1000 iterations. To speed up the optimization, GA stops if the average relative change in the best fitness value over 50 generations is less than or equal to 10^{-6}. To avoid getting stuck in local optima, 400 individuals are guaranteed to survive to the next iteration. The average run time is 121.8s.

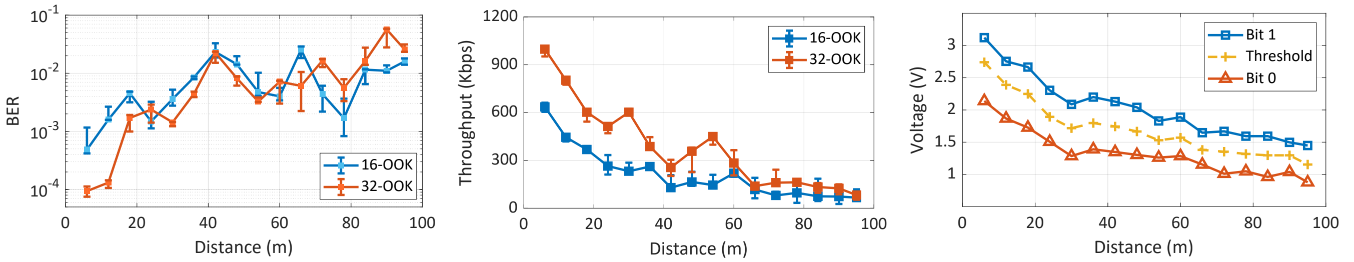
SA. SA has a simpler structure than GA. We designate the number of iterations as 6000 and the initial temperature as 63. The temperature at the n^{th} iteration is $TEMP^n = 0.997 \times TEMP^{n-1}$. To sustain exploration, we set a minimum temperature of 0.23. The average run time is 11.9s.

6.5 Compatible with COTS Wi-Fi Device

We patch the Nexmon [35] to the ASUS RT-AC86U Wi-Fi router [36] to manage the 802.11 transmission. Nexmon allows users to manually configure parameters such as the modulation and coding scheme (MCS) [37], channel coding scheme [38], scrambler initial value [39], and CP mode [37]. Specifically, we use 802.11n 20MHz channels with MCS to 7 (64-QAM, code rate 5/6) and a single spatial stream.

6.6 Hardware Prototype

Wook's RF front-end is identical to the wake-up receiver on the backscatter tag [4, 15, 16, 40–42] shown in Fig. 2. It consists of an envelope detector (ED), a comparator with an auto-threshold circuit, and a correlation demodulator. We use the ADL5513 envelope detector [43], with sensitivity tuned to -60 dBm, comparable to state-of-the-art ULP wake-up receiver ASICs. The TS3021 [44] serves as the comparator. The auto-threshold circuit is an RC integrator with $R = 1200\Omega$

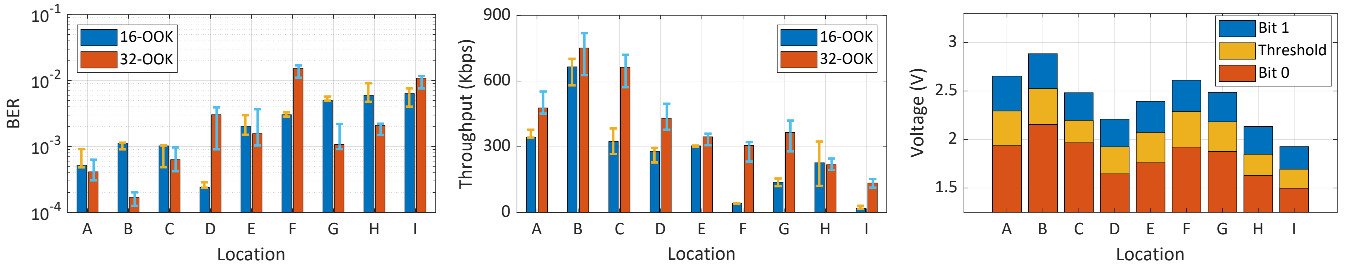


(a) BER performance.

(b) Throughput performance.

(c) Output voltage.

Figure 14: Performance of *Wook* in the line-of-sight scenario.



(a) BER performance.

(b) Throughput performance.

(c) Output voltage.

Figure 15: Performance of *Wook* in the non-line-of-sight scenario.

and $C = 33nF$. The low-power FPGA Gowin GW1N [45] is chosen as the correlation demodulator. The hardware prototype is shown in Fig. 11.

7 EVALUATION

We conduct experiments in an office building on a university campus, as illustrated in Fig. 10. The experiments are conducted in line-of-sight (LoS) and non-line-of-sight (NLoS) scenarios. The Wi-Fi AP is configured to send 802.11n signals on channel 9 (2452MHz). The primary performance metrics are throughput and BER.

7.1 Comparisons

Data rate. Figure 12 compares the achievable data rate of *Wook* with existing ULP Wi-Fi downlink systems. The packet-level modulation [10–12] achieves a data rate of less than 20 Kbps. Symbol-level modulation [13, 19] employs two OFDM symbols to convey one bit, resulting in a data rate of 125 Kbps [1]. *Wook* can modulate five bits on a single OFDM symbol, achieving a data rate of up to 1.1 Mbps, which is 55X and 8.8X higher than packet-level modulation and symbol-level modulation, respectively.

Communication distance. Our tests reveal that symbol-level modulation becomes ineffective for communication distances up to 6 m (BER up to 0.5). This ineffectiveness arises from the small energy difference between constant OFDM symbols and random OFDM symbols. Fig. 13 shows the energy difference distribution for 100 Wi-Fi frames at the same location, revealing an average energy difference of just 3.5 dB with symbol-level modulation. This small difference makes it difficult to accurately set the decision threshold for distinguishing the two energy states using passive auto-threshold circuits. In contrast, *Wook* employs heuristic algorithms to design Wi-Fi waveforms with an energy difference of up to 17.1 dB. Our experiments show that *Wook* achieves a BER of less than 3% at a distance of 95 m from the Wi-Fi router. To test the *Wook*'s extreme performance, we conduct an outdoor test, as shown in Fig. 16, where *Wook* demonstrate a maximum communication distance of up to 170 m, which is 28.3X that of symbol-level modulation.

7.2 *Wook* System Performance

LoS performance. The test results for the LoS scenario are shown in Fig. 14. *Wook* maintains a BER below 1% for communication distances up to 38 m, and even at 95 m, the BER stays under 4%. This impressive performance is due to two key factors: 1) *Wook* optimizes the Wi-Fi waveform to create a considerable energy disparity between the signal regions representing bit 0 and bit 1, enlarging the error margin for the decision threshold. 2) Adopting correlation decision instead of hard decision to demodulate OOK messages provides additional gains. Thanks to these enhancements, *Wook* consistently achieves a throughput of 255.5 Kbps with 32-OOK at distances up to 42 m. Even at 95 m, the throughput remains up to 82.9 Kbps. Notably, when the communication distance exceeds 70 m, 16-OOK has a lower BER compared to 32-OOK. Figure 14(c) shows the output voltage of the ED and auto-threshold circuits, demonstrating that the threshold consistently distinguishes between bit 1 and bit 0.

[1] The x-axis of Fig. 12 denotes GL [12], SS [11], HH [10], IS [13], WB [19].

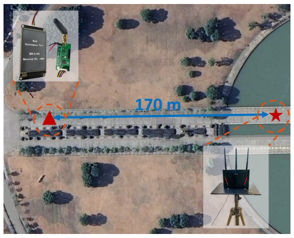
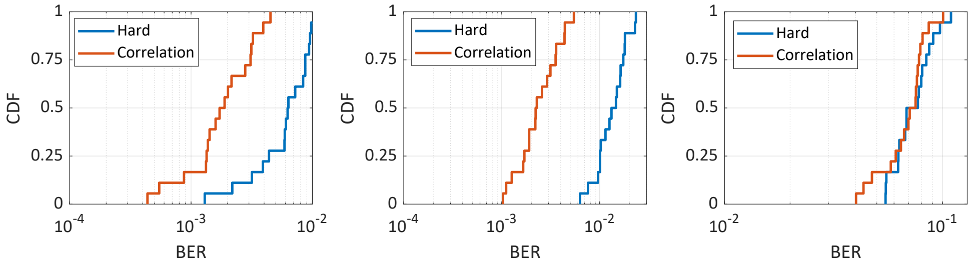


Figure 16: Maximum communication distance test.



(a) 30 m

(b) 60 m

(c) 90 m

Figure 17: Performance gap between hard decision and correlation decision.

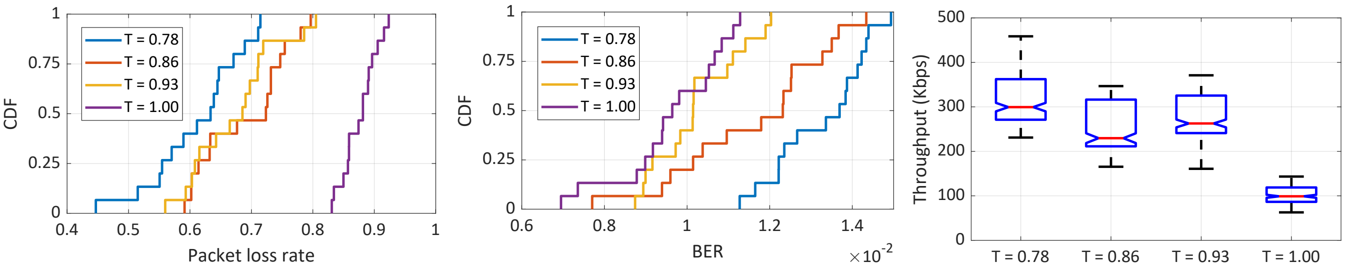


Figure 18: Performance variation at different wake-up thresholds.

NLoS performance. Next, we evaluate the *Wook*'s NLoS performance across an area of 180m^2, as shown in Fig.10(b). The detailed results are presented in Fig. 15. Test points A, B, and C are situated behind a wall and consistently exhibit a BER below 10^{-3}. The BER always remains below 2% in all the test points. Figure 15(b) presents the throughput of *Wook* in NLoS scenarios. With 32-OOK, the maximum throughput is 750.8 Kbps at point B and a minimum of 134.6 Kbps at point I. These detailed evaluations affirm that *Wook* operates reliably under diverse environmental conditions.

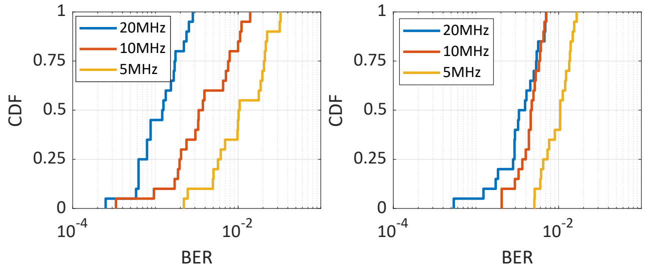
Comparing the LoS and NLoS test results, we observe that the BER performance difference between 32-OOK and 16-OOK is minimal, even though the duration of each bit in 32-OOK is 0.15μs less. *This suggests that Wook has substantial potential for realizing higher-order OOK modulations.*

7.3 Micro Benchmarks

Demodulation method. In Sec. 5.3, we introduce a correlation decision method. Here, we evaluate its BER performance across various distances and compare it to the hard decision, which demodulates OOK bits individually using a simple voting method. The results in Fig. 17 highlight the superiority of correlation decision, showing significant BER reductions by factors of 3.3 and 5.8 at distances of 30 m and 60 m, respectively. This improvement stems from the correlation decision's ability to leverage samples captured over the entire OOK symbol, increasing the available data volume and bringing additional gains. The gains become less apparent as the communication distance increases to 90 m. This is due to the weakened received signal and the accumulation of too many erroneous samples.

Wake-up threshold. To synchronize with the Wi-Fi AP, *Wook* uses a 16-bit frame header as the wake-up sequence. *Wook* calculates the correlation coefficient of the sampled bit sequence with this wake-up sequence. Once the correlation coefficient surpasses a predetermined threshold, *Wook* starts to receive the OOK payload following the wake-up sequence. A lower threshold facilitates the reception of downlink signals, thus increasing the system's throughput. However, it also makes the system more susceptible to co-channel interference, which can raise the BER. Figure 18 illustrates the impact of different thresholds on the packet loss rate, BER, and throughput when the Wi-Fi AP is 40 m away from the *Wook* device. A lower threshold results in a lower packet loss rate, while the difference is negligible when the threshold is set to 0.86 or 0.93. Conversely, the packet loss rate increases significantly when the threshold exceeds 0.93. Figure 18(b) indicates that a higher threshold corresponds to a lower BER, with minimal difference observed between thresholds of 0.93 and 1. The throughput variation shown in Fig. 18(c) follows a similar pattern. Therefore, to balance the throughput and BER, *Wook* sets the threshold to 0.93.

Sampling rate. The sampling rate of the comparator's output significantly impacts system performance. We conduct BER tests at different locations using various sampling rates, and the results are shown in Fig. 19. It is evident that higher sampling rates lead to lower BERs. For example, a 20 MHz sampling rate at test point A reduces the BER by 3 and 8.32 compared to 10 MHz and 5 MHz, respectively. At more distant test point G, the benefit of a 20 MHz sampling rate over 10 MHz is less pronounced. This observation is



(a) BER performance in position A. (b) BER performance in position G.

Figure 19: Performance with different sampling rates.

crucial for ULP uplink systems like Wi-Fi backscatter, where reflected signals are frequency-shifted by at least 20 MHz to avoid conflict with the excitation signals. Therefore, *Wook*'s use of a 20 MHz sampling rate does not incur any additional hardware overhead for existing ULP devices.

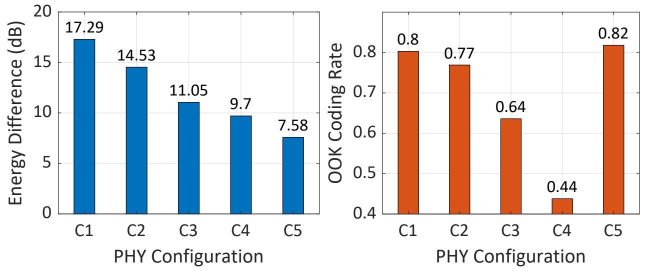
Wi-Fi PHY configurations. We evaluate how the channel bandwidths and MCS effect *Wook*. The results are shown in Fig. 20(a). Interestingly, increasing the QAM order or using larger bandwidths expands the solution space but results in a lower energy difference. This occurs because the exponential expansion of the solution space makes it difficult for heuristic algorithms to explore fully within a limited time. Conversely, 16-QAM and 4-QAM are in the opposite situation. Fig. 20(b) presents the OOK coding rate under different configurations. Although QPSK offers a slightly higher coding rate than 64-QAM, its energy difference is too small to meet *Wook*'s requirements. Taking these factors into account, *Wook* is implemented with a 20MHz/64-QAM configuration.

7.4 IC power consumption

The IC design of *Wook* involves analog and digital parts. 1) For the analog parts, Cadence Virtuoso [46] with SMIC 40 nm process is used. A 20 MHz ring oscillator to sample the comparator output, utilizing the same architecture as in [47], with a power consumption of 19.5μW; The envelope detector and comparator together consume 53.7μW. 2) For the digital parts, Synopsys DC with SMIC 40nm library [48] is utilized to synthesize and report power consumption under a 1.1 V voltage supply. The wake-up circuit demodulates the OOK signals using hard decisions to minimize resource usage and reduce power consumption, consuming 0.8μW. The correlation demodulator primarily employs multiplexers to compare the correlation coefficients with a power consumption of 3.4μW. Additionally, *Wook* requires RAM to store the wake-up pattern and the OOK symbol templates, with a power consumption of 0.2μW. Thus, the overall power consumption of *Wook* is 77.6μW.

8 RELATED WORK

ULP downlink systems. 1) Wi-Fi. HitchHike *et al.* [10, 11, 18] use packet-level modulation to convey OOK bits. However, the duration of a Wi-Fi packet is usually a few tens of microseconds, so packet-level modulation communication is only capable of tens of Kbps. Interscatter [13] employ two OFDM symbols to convey one OOK bit, thus achieving a data rate of 125Kbps. [12] modifies the RF ports of Wi-Fi devices by introducing additional hardware to change the energy strength of the transmitted signals with a communication rate of less than 10 Kbps. MIXIQ [29] achieves Mbps data rates, but it only works with 802.11ax and is incompatible with the COTS Wi-Fi router. 2) LoRa. Saiyan [14] enables low-power LoRa signal demodulation for ULP devices to realize long-distance downlink data transmission. 3) Custom protocols. Passive DSSS [15] designs interference-resistant ULP downlink with DSSS. Analog OFDMA [17] achieves highly concurrent downlink communication by filtering out subcarriers. µmote [16] achieves long-range communication based on chirp modulation.



(a) Energy difference. (b) Encoding rate.

Figure 20: Performance with different Wi-Fi PHY configurations. C1: 20M/64QAM, C2: 20M/16QAM, C3: 40M/256QAM, C4: 40M/64QAM, C5: 20M/4QAM

ULP downlink hardware. The development and application of integrated circuits to design ultra-low power, high-sensitivity OOK receivers have become increasingly prevalent. [49] achieves -42.6 dBm sensitivity at 2.8μW. [50] achieves -50 dBm sensitivity at 4.5μW. [51] achieves -51 dBm sensitivity with 10μW. [52] achieves -64 dBm sensitivity at 51μW. [53] achieves -71 dBm sensitivity at 11μW. [54] achieves -72 dBm sensitivity at 93μW.

Cross-technology communication (CTC). *Wook* converts Wi-Fi signals into high-speed OOK signals, categorizing it a type of CTC. Traditional Wi-Fi based CTC relies on 802.11 reverse engineering to find Wi-Fi payloads that emulate the desired waveform. Examples include WeBee [19], LongBee [55], BlueFi [20] and WiRa [56]. However, reverse engineering faces the ill-pose problem, wherein the desired output of the BCC encoder may not constitute a valid codeword, resulting in rough emulations and poor performance [57]. The use of LDPC allows *Wook* to perform 802.11 forward analysis, thus solving the ill-pose problem. Another emerging approach uses AI for global optimization. XiTuXi [57] views the system as a black box and employs popular transformer to ease the design efforts, while NNCTC [58] uses neural networks with fewer training samples. However, these methods are computationally intensive and impractical for real-time communication. For example, for each emulation, even on high-end GPUs/CPUs (e.g., RTX 3080Ti or i9-12900H), XiTuXi takes about 1s, and NNCTC takes a staggering 3 minutes. In contrast, *Wook* acts as a "white box," necessitating merely a simple lookup table, which facilitates rapid processing. For example, transmitting 256 OOK symbols (equivalent to use 256 OFDM symbols to convey a ZigBee frame in XiTuXi and NNCTC), *Wook* with a 5 GHz (i9-12900H max frequency) clock only requires $\frac{256}{5GHz} = 51.2ns$.

ULP uplink systems. 1) Wi-Fi. [59] realizes in-band backscatter communication achieves a throughput of 248kbps. [10] utilizes 802.11b and achieves a throughput of 300 Kbps at a distance of 1 m. [5] employs a multi-antenna technique to achieve a throughput of 500 Kbps. [11] implements a two-stage wake-up strategy, boosting the throughput to 600 Kbps. [7] introduces sub-symbol level modulation, achieving a throughput of 743 Kbps. [9] synthesizes Wi-Fi signals using pure carriers, achieving a throughput of 1 Mbps. [6] reaches a throughput of about 1.1 Mbps with 16-PSK modulation. [3] even boosts throughput to a remarkable 10 Mbps. These works achieve throughput significantly higher than the state-of-the-art Wi-Fi ULP downlink systems. 2) Other protocols. LoRa backscatter [60–62] enables long-distance communication. LTE backscatter [41, 63] enables uninterrupted data transmission using ubiquitous LTE signals. [64] enables the active transmission of BLE signals without needing a battery.

9 DISCUSSION AND FUTURE WORK

Influence of pilots. In the waveform optimization process, we assume the pilots are know. However, 802.11 assigns different pilots to each OFDM symbol. We observe the following: 1) the pilots for all OFDM symbols are known once the Wi-Fi PHY parameters (frame length, channel bandwidth etc.) are configured, and 2) there are eight pilot combinations in 802.11n 20MHz channels. Thus, *Wook* gives eight solutions for each OOK symbol. This operation has minimal overhead. For example, with 32-OOK, only $\frac{32\times8\times52\times6}{1024\times8} = 9.75KB$ of memory is needed to store the *Wook*'s encoding messages. This requirement is negligible for most COTS Wi-Fi routers, e.g., the RT-AC86U has a memory size of 256MB.

High order OOK modulation. According to the results in Sec. 7.2, the performance gap between 16-OOK and 32-OOK is minimal. This indicates that *Wook* has the potential to employ higher-order OOK modulations, such as 64-OOK or even 256-OOK to further improve the throughput. However, implementing higher-order modulation requires substantial effort to optimize the waveforms. Therefore, designing a more efficient optimization algorithm is our next goal.

802.11ax/be. Compared to 802.11n/ac, the OFDM symbol duration (excluding the CP) has increased from $3.2\mu s$ to $12.8\mu s$ in 802.11ax and 802.11be. As a result, 802.11ax/be-*Wook* requires a 4X in OOK modulation order to achieve a throughput comparable to 802.11n/ac-*Wook*, demanding 4X more efforts. Wi-Fi standards are backward compatible, so even the latest 802.11be devices will continue support 802.11n/ac. This ensures *Wook* to be easily integrated into existing COTS Wi-Fi devices.

5G NR low-power wake-up signal (LPWUS). 3GPP conducts a study on LPWUS [65], proposing four OOK-based options (OOK-3 works in the frequency domain like OOK-2, is omitted here). OOK-1 resembles symbol-level modulation in Wi-Fi downlink, where all samples in the LPWUS field convey a single bit. OOK-2 transmits multiple OOK bits using separated subcarrier segments, requiring receivers to employ parallel band-pass filters to isolate each segment. This demands precise filter design, such as high-Q components to narrow filter bandwidth (a hardware example shown in [17]). Moreover, the circuit size increases with the modulation order. OOK-4 operates in the time domain, but its waveform generation remains under exploration, and the system design lacks practical deployment. *Wook* offers a design paradigm for high-speed OOK waveforms in OFDM environments, providing valuable insights for LPWUS design in NR.

Effects on normal Wi-Fi links. Our evaluation platform based on Nexmon. The activating Nexmon disables the AP's wireless connectivity. However, a simple firmware update from the Wi-Fi chip manufacturer — storing the optimized payload in Sec.5.2 in memory and configuring PHY parameters from Sec.6.5 — is sufficient to deploy *Wook* on COTS Wi-Fi devices. This requires no modifications to the chips or the 802.11 standards. Once deployed, the *Wook* device functions as a regular Wi-Fi terminal and does not interfere with the normal Wi-Fi link.

10 CONCLUSIONS

We introduce *Wook*, a novel system that enables ultra-low power downlink data transmission in Wi-Fi environments. Leveraging heuristic algorithms and a comprehensive analysis of the 802.11 transmission flow, *Wook* achieves sub-symbol modulation by manipulating the Wi-Fi waveform. Our results illustrate that *Wook* achieves a throughput of up to 1.1 Mbps, which surpasses state-of-the-art approaches by a factor of 8.9. This innovative design empowers ultra-low power devices to communicate with existing wireless infrastructure with high throughput, thus unlocking the potential of future IoT networks.

ACKNOWLEDGEMENTS

The work in this paper is supported by the National Key Research and Development Program of China 2020YFB1708700, and National Natural Science Foundation of China (No. 61922055, 61872233).

REFERENCES

[1] Wi-Fi IoT Market Report: Trends, Forecast and Competitive Analysis to 2030. https://www.researchandmarkets.com/reports/5928857/wi-fi-iot-market-report-trends-forecast.

[2] R. Zhao, K. Wang, K. Zheng, X. Zhang, and V. Leung, “SlimWiFi: Ultra-Low-Power IoT radio architecture enabled by asymmetric communication,” in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, (Boston, MA), pp. 1201–1219, USENIX Association, Apr. 2023.

[3] Q. Qin, K. Chen, Y. Xie, H. Luo, D. Fang, and X. Chen, “Pushing the throughput limit of ofdm-based wi-fi backscatter communication,” in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom ’24, (New York, NY, USA), p. 968–983, Association for Computing Machinery, 2024.

[4] X. Liu, Z. Chi, W. Wang, Y. Yao, P. Hao, and T. Zhu, “Verification and redesign of OFDM backscatter,” in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pp. 939–953, USENIX Association, Apr. 2021.

[5] X. Liu, Z. Chi, W. Wang, Y. Yao, and T. Zhu, “{VMscatter}: A versatile {MIMO} backscatter,” in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pp. 895–909, 2020.

[6] Y. Yang, L. Yuan, J. Zhao, and W. Gong, “Content-agnostic backscatter from thin air,” in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pp. 343–356, 2022.

[7] J. Yu, C. Du, J. Liu, R. Zhang, and S. Wang, “Subscatter: Subcarrier-level ofdm backscatter,” in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pp. 1–10, IEEE, 2023.

[8] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti, “Backfi: High throughput wifi backscatter,” *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 283–296, 2015.

[9] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith, “Passive {Wi-Fi}: Bringing low power to {Wi-Fi} transmissions,” in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pp. 151–164, 2016.

[10] P. Zhang, D. Bharadia, K. Joshi, and S. Katti, “Hitchhike: Practical backscatter using commodity wifi,” in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pp. 259–271, 2016.

[11] M. Dunna, M. Meng, P.-H. Wang, C. Zhang, P. Mercier, and D. Bharadia, “Syncscatter: Enabling wifi like synchronization and range for wifi backscatter communication,” in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pp. 923–937, 2021.

[12] Z. Kapetanovic, A. Saffari, R. Chandra, and J. R. Smith, “Glaze: Overlaying occupied spectrum with downlink iot transmissions,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, pp. 1–21, 2019.

[13] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith, “Inter-technology backscatter: Towards internet connectivity for implanted devices,” in *Proceedings of the 2016 ACM SIGCOMM Conference*, pp. 356–369, 2016.

[14] X. Guo, L. Shangguan, Y. He, N. Jing, J. Zhang, H. Jiang, and Y. Liu, “Saiyan: Design and implementation of a low-power demodulator for LoRa backscatter systems,” in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, (Renton, WA), pp. 437–451, USENIX Association, Apr. 2022.

[15] S. Li, H. Zheng, C. Zhang, Y. Song, S. Yang, M. Chen, L. Lu, and M. Li, “Passive DSSS: Empowering the downlink communication for backscatter systems,” in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, (Renton, WA), pp. 913–928, USENIX Association, Apr. 2022.

[16] Y. Song, L. Lu, J. Wang, C. Zhang, H. Zheng, S. Yang, J. Han, and J. Li, “{μMote}: Enabling passive chirp de-spreading and {μW-level}{Long-Range} downlink for backscatter devices,” in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pp. 1751–1766, 2023.

[17] F. Zhu, L. Feng, M. Jin, X. Tian, X. Wang, and C. Zhou, “Towards ultra-low power ofdma downlink demodulation,” in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, SenSys ’22, (New York, NY, USA), p. 725–739, Association for Computing Machinery, 2023.

[18] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, “Wi-fi backscatter: Internet connectivity for rf-powered devices,” in *Proceedings of the 2014 ACM Conference on SIGCOMM*, pp. 607–618, 2014.

[19] Z. Li and T. He, “Webee: Physical-layer cross-technology communication via emulation,” in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, MobiCom ’17, (New York, NY, USA), p. 2–14, Association for Computing Machinery, 2017.

[20] H.-W. Cho and K. G. Shin, “Bluefi: bluetooth over wifi,” in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, SIGCOMM ’21, (New York, NY, USA), p. 475–487, Association for Computing Machinery, 2021.

[21] J. G. Proakis and M. Salehi, *Digital communications.* McGraw-hill, 2008.

[22] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, and S. Han, “Tiny machine learning: Progress and futures [feature],” *IEEE Circuits and Systems Magazine*, vol. 23, no. 3, pp. 8–34, 2023.

[23] K. Kopparapu, E. Lin, J. G. Breslin, and B. Sudharsan, “Tinyfedtl: Federated transfer learning on ubiquitous tiny iot devices,” in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 79–81, 2022.

[24] Y. Li, D. Qin, H. V. Poor, and Y. Wang, “Introducing edge intelligence to smart meters via federated split learning,” *Nature Communications*, vol. 21, no. 1, 2024.

[25] W. Badawy, A. Ahmed, S. Sharf, R. A. Elhamied, M. Mekky, and M. A. Elhamied, “On flashing over the air “fota” for iot appliances – an atmel prototype,” in *2020 IEEE 10th International Conference on Consumer Electronics (ICCE-Berlin)*, pp. 1–5, 2020.

[26] R. Sanchez-Iborra, “Lpwan and embedded machine learning as enablers for the next generation of wearable devices,” *Sensors*, vol. 21, no. 15, 2021.

[27] N. Schizas, A. Karras, C. Karras, and S. Sioutas, “Tinyml for ultra-low power ai and large scale iot deployments: A systematic review,” *Future Internet*, vol. 14, no. 12, 2022.

[28] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, Jan. 2019.

[29] M. Rostami, X. Chen, Y. Feng, K. Sundaresan, and D. Ganesan, “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*, pp. 364–377, 2021.

[30] Y. Fang, G. Bi, Y. L. Guan, and F. C. M. Lau, “A survey on protograph ldpc codes and their applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1989–2016, 2015.

[31] M. Tawarmalani and N. V. Sahinidis, “Global optimization of mixed-integer nonlinear programs: A theoretical and computational study,” *Mathematical programming*, vol. 99, no. 3, pp. 563–591, 2004.

[32] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C (2nd ed.): the art of scientific computing.* USA: Cambridge University Press, 1992.

[33] M. Srinivas and L. M. Patnaik, “Genetic algorithms: A survey,” *computer*, vol. 27, no. 6, pp. 17–26, 1994.

[34] P. Zhang, C. Josephson, D. Bharadia, and S. Katti, “Freerider: Backscatter communication using commodity radios,” in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pp. 389–401, 2017.

[35] M. Schulz, D. Wegemer, and M. Hollick, “Nexmon: The c-based firmware patching framework,” 2017. https://nexmon.org.

[36] ASUS RT-AC86U WiFi Routers, https://www.asus.com/us/networking-iot-servers/wifi-routers/asus-wifi-routers/rt-ac86u/.

[37] “Exampls to set mcs and cp mode..” tx_task.sh, https://github.com/seemoo-lab/nexmon_tx_task/blob/main/utils/tx_task.sh#L78.

[38] “Tutorials to set channel coding scheme..” Issue#5, https://github.com/seemoo-lab/nexmon_tx_task/issues/5.

[39] “Tutorials to set scrambler’s initial value..” Issue#7, https://github.com/seemoo-lab/nexmon_tx_task/issues/7.

[40] R. Zhao, F. Zhu, Y. Feng, S. Peng, X. Tian, H. Yu, and X. Wang, “Ofdma-enabled wi-fi backscatter,” in *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom ’19, (New York, NY, USA), Association for Computing Machinery, 2019.

[41] Z. Chi, X. Liu, W. Wang, Y. Yao, and T. Zhu, “Leveraging ambient lte traffic for ubiquitous passive communication,” in *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM ’20, (New York, NY, USA), p. 172–185, Association for Computing Machinery, 2020.

[42] B. Wang, F. Zhu, W. Li, Z. Yang, M. Jin, and X. Tian, “Frequency-agile ofdm backscatter,” in *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, p. 252–264, ACM New York, NY, USA, 2024.

[43] ADL5513, https://www.analog.com/En/products/adl5513.html.

[44] TS3201, https://www.st.com/en/automotive-analog-and-power/ts3021.html.

[45] GW1N-LV4QN48C6/I5. https://www.gowinsemi.com/en/product/detail/46/.

[46] Virtuoso Layout Suite. https://www.cadence.com/ko_KR/home/tools/custom-ic-analog-rf-design/layout-design/virtuoso-layout-suite.html.

[47] P. Zhang, M. Rostami, P. Hu, and D. Ganesan, “Enabling practical backscatter communication for on-body sensors,” in *Proceedings of the 2016 ACM SIGCOMM Conference*, pp. 370–383, 2016.

[48] DC Ultra. https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/dc-ultra.html.

[49] P.-H. P. Wang, C. Zhang, H. Yang, D. Bharadia, and P. P. Mercier, “20.1 a 28μw iot tag that can communicate with commodity wifi transceivers via a single-side-band qpsk backscatter communication technique,” in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 312–314, IEEE, 2020.

[50] S.-E. Chen, C.-L. Yang, and K.-W. Cheng, “A 4.5 μw 2.4 ghz wake-up receiver based on complementary current-reuse rf detector,” in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1214–1217, IEEE, 2015.

[51] J. Shen, F. Zhu, Y. Liu, B. Liu, C. Shi, L. Huang, L. Xu, X. Tian, and R. Zhang, “23.1 a 44μw iot tag enabling 1μs synchronization accuracy and ofdma concurrent communication with software-defined modulation,” in *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 67, pp. 400–402, IEEE, 2024.

[52] X. Huang, S. Rampu, X. Wang, G. Dolmans, and H. de Groot, “A 2.4 ghz/915mhz 51μw wake-up receiver with offset and noise suppression,” in *2010 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 222–223, IEEE, 2010.

[53] K.-W. Cheng and S.-E. Chen, “An ultralow-power ook/bfsk/dbpsk wake-up receiver based on injection-locked oscillator,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 7, pp. 1379–1391, 2021.

[54] E. Alpman, A. Khairi, R. Dorrance, M. Park, V. S. Somayazulu, J. R. Foerster, A. Ravi, J. Paramesh, and S. Pellerano, “802.11 g/n compliant fully integrated wake-up receiver with- 72-dbm sensitivity in 14-nm finfet cmos,” *IEEE Journal of Solid-State Circuits*, vol. 53, no. 5, pp. 1411–1422, 2018.

[55] Z. Li and T. He, “Longbee: Enabling long-range cross-technology communication,” in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 162–170, 2018.

[56] X. Zheng, D. Xia, F. Yu, L. Liu, and H. Ma, “Enabling cross-technology communication from wifi to lora with ieee 802.11ax,” *IEEE/ACM Transactions on Networking*, vol. 32, no. 3, pp. 1936–1950, 2024.

[57] S. Liao, Z. An, Q. Pan, X. Zhao, J. Tong, and L. Yang, “Xituxi: Sealing the gaps in cross-technology communication by neural machine translation,” in *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, SenSys ’23, (New York, NY, USA), p. 70–82, Association for Computing Machinery, 2024.

[58] H. Wang, J. Wang, D. Gao, and W. Jiang, “Nnctc: Physical layer cross-technology communication via neural networks,” in *2024 23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 51–62, 2024.

[59] C. Du, J. Yu, R. Zhang, J. Ren, and J. An, “Orthcatter: High-throughput in-band OFDM backscatter with Over-the-Air code division,” in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, (Santa Clara, CA), pp. 1301–1314, USENIX Association, Apr. 2024.

[60] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, “Lora backscatter: Enabling the vision of ubiquitous connectivity,” *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 1, no. 3, pp. 1–24, 2017.

[61] J. Jiang, Z. Xu, F. Dang, and J. Wang, “Long-range ambient lora backscatter with parallel decoding,” in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pp. 684–696, 2021.

[62] M. Katanbaf, A. Weinand, and V. Talla, “Simplifying backscatter deployment:{Full-Duplex}{LoRa} backscatter,” in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pp. 955–972, 2021.

[63] W. Li, M. Jin, and X. Tian, “Intra-band lte backscatter,” *Journal of Communications and Information Networks*, vol. 9, no. 3, pp. 207–218, 2024.

[64] J. de Winkel, H. Tang, and P. Pawełczak, “Intermittently-powered bluetooth that works,” in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, MobiSys ’22, (New York, NY, USA), p. 287–301, Association for Computing Machinery, 2022.

[65] 3GPP TR 38.869. https://www.3gpp.org/ftp/Specs/archive/38_series/38.869/.