Yang *et al. Journal of Cloud Computing* (2023) 12:157
https://doi.org/10.1186/s13677-023-00508-5

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

NuWa: off-state tolerant backscattering system with uncontrolled excitation traffics

Zhiyi Yang[1], Xin He[1,2*], Guiping Lin[3], Weiwei Jiang[2,4*], Yujun Zhu[1], Jianfeng Sun[1], Yong Xu[1] and Panlong Yang[2,5]

Abstract

Backscatter communication relies on passive reflections of the existing radio frequency (RF) signals, making it suitable for low-power and low-complexity communication in IoT applications. However, the performance of existing systems severely degrades in real-life environments, due to irregular "on" and "off" states of ambient signals like WiFi, which are not controllable. In this paper, we propose a joint coding and framing scheme for the backscattering physical layer to fight against the off-state in the excitation signal. We first design transmission schemes including both the Reed-Solomon (RS) codes and the frame structure, to correct the burst error caused by the off states. In order to implement the codes at the resource-constrained tag, we design a look-up table for the encoding process. We prototype our system NuWa that could efficiently backscatter with uncontrolled traffics generated randomly. We demonstrate that NuWa could achieve a 1 Mbps transmission throughput when the tag is over 1 *m* away from the receiver in high traffic load and 150 kbps in low traffic load. Finally, we evaluate the throughput with respect to the distance change between the tag and the receiver, and 950 kbps is achieved at a distance of 6 *m*.

Keywords WiFi backscatter, Uncontrolled traffic, Reed-Solomon code, Frame design

Introduction

Mobile devices powering the Internet of Things (IoT) are ubiquitous nowadays, with an increasing number of IoT devices deployed in our surrounding environments. One notable trend is that these IoT devices become even smaller. With a very small size, powering these devices becomes challenging [1]. In particular, attaching to a power cord is not always practical or feasible for deployment such as in the wild, while extra batteries add weight, size and cost. One promising solution that has been proposed in recent years, is backscatter communication [2, 3]. It leverages existing ambient RF traffics, rather than emitting its own radio signals, which greatly reduces the amount of power consumption. In consequence, backscatter communication is orders of magnitude more power-efficient than traditional radio technologies such as WiFi or 4G LTE. Further, since it takes advantage of the ambient RF signals that already exist, it does not require a dedicated power infrastructure such as the reader for RFID.

In particular, various RF technologies have been used as the excitation sources in existing works for backscatter communications, including TV [2], FM [4] and WiFi [2, 5]. Among these excitation sources, WiFi is considered particularly promising due to its ubiquity. Exciting progress has been achieved with WiFi backscattering and the state-of-the-art achievable throughout is 5 Mbps at a transmission distance of 1 meter [6], capable of supporting lots of real-life applications. However, there are some practical concerns to deploying the WiFi backscatter systems, such as signal synchronization, strong

*Correspondence:
Xin He
xin.he@ahnu.edu.cn
Weiwei Jiang
weiwei.jiang@ahnu.edu.cn
[1] The AIoT Lab, Anhui Normal University, Wuhu, China
[2] School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, China
[3] School of Electronics and Information Engineering, Harbin Institute of Technology, Shenzhen, China
[4] The University of Melbourne, Melbourne, Australia
[5] School of Computer Science and Technology, Nanjing University of Information Science and Technology, Nanjing, China

interference from the legacy channel and the intermittent manner (burst transmission) of WiFi signals. According to the theoretical analysis in [7], the intermittent nature strongly affects the backscatter system performance. Hence, we focus on solving the low reliability of the backscattering link caused by the intermittent WiFi excitation.

Existing WiFi backscattering schemes heavily rely on fully controlled WiFi traffic to achieve a notable performance. For example, in the lab environment, researchers can fully control the WiFi access point (AP) to achieve good results, *i.e.*, it is able to inject sufficient dummy packets in the WiFi channel to support the backscatter link. However, the performance significantly degrades in real-life with uncontrolled WiFi traffic [7]. In reality, it is very challenging to control the publicly available WiFi AP, as it may greatly affect the ongoing WiFi data communication, or involves security and privacy concerns.

Therefore, in this paper, we aim to solve the problem when the backscatter tag encounters the off-state in the WiFi excitation signals, for moving one step toward real-life deployment of WiFi backscattering systems. To achieve this, our main objective is to make WiFi backscattering flexibly and efficiently work with uncontrolled WiFi traffic from widely available "roadside APs"[1]. Multiple challenges need to be tackled before we can realize this ambitious objective.

The first challenge is that real-life WiFi traffic is unpredictable in terms of packet length and packet interval [8, 9]. So the traffic "on" and "off" states are totally out of control. On the other hand, for backscattering, it is better to have the information about when the traffic (excitation source) is "on" and how long the "on" state will be. Without knowing the "on/off" states, backscattering becomes extremely difficult.

The second challenge is that the tag in the backscatter communication system is resource-constrained in terms of both power and computational resources. Therefore, the design needs to take care of both the computational complexity and power consumption.

To address the first challenge, we propose a novel solution based on coding theory. The key observation that motivates our solution is that although the on/off state is hard to predict, the interval between the on-off states is usually short in the scale of 1000 μs based on the IEEE 802.11 standard. This is especially true when the traffic load is high. This observation motivates us to backscatter without the need of knowing the on/off states. Although burst errors occur during the off state, as long as the off state is not too long, we can still design a smart coding scheme to deal with the burst error and recover the data.

[1] We name those APs in the environment which are not under our control as "roadside APs"

To address the second challenge, we design an energy-efficient Reed-Solomon code that works well with ultra-low power hardware, of which the concept is shown in Fig. 1.

To summarize, the contributions of this work are as below.

- In tackling the "on-off" states variation, we advocate leveraging the small interval of the "off state" for transmission and regard the data in these intervals as burst errors. With properly designed schemes, these burst errors can be effectively reduced.
- Built upon this design vision, we further propose to tame the uncontrollable WiFi APs at the roadside by a joint design of the backscatter frame structure and coding scheme. Especially, our design is power and complexity efficient which can work with resource-constrained low-cost hardware in real-time.
- We self-design the PCB for the tag and employ the ultra-low power microcontroller MSP432P401R to control the tag's operations. We prototype NuWa, a working system that could efficiently utilize the random excitation for backscattering, and the hardware platform can connect sensors through a generic GPIO pin.

Related work

NuWa is a WiFi backscatter system that firstly aims to improve the reliability of backscatter communication over WiFi signals from the roadside APs. We use the statistical knowledge about the alternation of on-and-off states of the WiFi signal in the wild and optimize an RS code to follow the transition of WiFi states. The WiFi backscatter system has been intensively studied recently in [5, 6, 10–16].

Backscatter communication

Thanks to the great efforts by the research community to enable backscatter communication over various

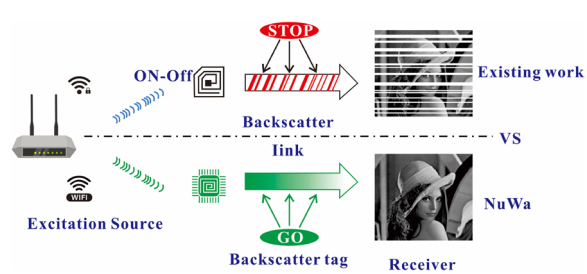


Fig. 1 NuWa enables WiFi backscattering using the Roadside WiFi APs as the excitation sources. The backscatter tag in NuWa actively adjusts the backscatter schemes to fit the variations of WiFi signals

radio signals. A breakthrough work of ambient backscatter communication [2] built a prototype which can backscatter information through TV signals. LoRa backscatter [17] and PLoRa [18] enabled long-range backscatter communication using LoRa signal as excitation. FM backscatter [4] adopted continuous FM radio signal to act as the excitation and can enable several new applications.

WiFi backscatter

BackFi [6] operated backscatter communication over the WiFi excitation signals transmitted from WiFi APs with hardware modification. Reference [5] connected the RF-powered devices to the Internet excited by a WiFi signal. Passive WiFi demonstrates for the first time it is able to generate 802.11b backscatter transmissions using backscatter communications [11]. HitchHike [12] and [14] enabled the backscatter communication over 802.11b signals of the COTS WiFi transceivers using the proposed codeword translation technique. FreeRider [10] further extends the backscatter communication over other excited RF radios, such as Bluetooth, 802.11g/n WiFi and ZigBee. Recent work [13] enables per-symbol and in-band backscatter communication over the WiFi excitation signals using a so-called flicker detector by utilizing the residual channel knowledge of the WiFi packets. Besides, backscattering the ultra-wideband signals is considered in [19]. X-Tandem [15] enabled multi-hop backscatter systems with WiFi signals as excitation sources. However, how to use uncontrolled WiFi traffic as an excitation signal is rarely considered in these systems. In order to enable flexible backscatter communication, the inherent nature of the excitation signal should be considered. In [7], the authors analyzed the capacity of a backscatter system with random excitation following the Bernoulli distribution, where the results verify that the system performance is dominated by the random nature of the excitation. GuardRider [20] system made a try to backscatter information in the real WiFi networks. However, it simply assumed the WiFi duration follows Pareto distributions without verifying the correctness by the real WiFi traffic. RapidRider [21] system embedded the tag's data into a single OFDM symbol to reduce the effect of the uncontrolled WiFi traffic. We further adopted low-density parity-check (LDPC) codes and designed a rateless LDPC to overcome the intermittent nature of the excitation signals [22].

Preliminary knowledge

A brief review of WiFi protocols

WiFi signals can be found almost everywhere in urban areas, including both outdoors and indoors. Typically, WiFi networks operate in 2.4 GHz and 5 GHz frequency bands which adopt the carrier sense multiple access with collision avoidance protocol to decrease the collision chances when sharing the same transmission medium [23], *i.e.*, the free space.

WiFi protocol indeed employs the inter-frame space (IFS) (*e.g.*, DIFS and SIFS) to show how much delay the WiFi stations should have before transmission [2]. By sensing the channel medium, a WiFi station determines whether there is another station transmitting. Provided the medium is busy, the station will then defer. If the medium is idle during the duration of DIFS, the station attempts to transmit. As the receiver receives the frame in the correct way and verifies the frame using cyclic redundancy check (CRC), it sends an ACK back to the sender. If the sender does not receive the ACK, it will retransmit the frame until it gets ACK or the number of retransmissions exceeds the maximum allowed. To address the well-known hidden terminal issue in wireless network scenarios, 802.11 further applies short control frames (like RTS and CTS) before transmitting the data packet, to inform the other stations.

Exponential backoff is adopted in 802.11 protocols. Following the DIFS, there is a contention divided into slots within which stations select a random timer value at random and count down until that timer expires before attempting to transmit data. The contention window is initialized at a minimum size of 31 slots and will be doubled when it fails to access the medium (exponential backoff) until it reaches the maximum of 1023 slots[3].

WiFi station usually transmits packets with a small size due to: (1) A packet encounters a higher possibility if its packet size gets larger; (2) If the packet gets corrupted, the station spends more overhead in retransmitting a longer packet. Therefore, in WiFi networks, the station usually chops the long packets, for instance, the frame in an Ethernet, into several fragments. Therefore, the WiFi frame duration is indeed short [9].

Backscatter communications and WiFi backscatter

A typical backscattering communication system is depicted in Fig. 2, which is composed of an excitation source, a backscatter tag and a receiver.

Rather than generating the carrier signal itself, the backscatter tag in the backscatter system modulates its data on the ambient wireless signal by adjusting the impedance of its antenna. The received power from the backscatter tag is denoted as

[2] WiFi station and WiFi AP are alternatively used in this paper.

[3] Typically, each slot lasts for 20 μs, or 9 μs if there are no direct-sequence spread spectrum (DSSS) stations in 802.11 g/n networks.

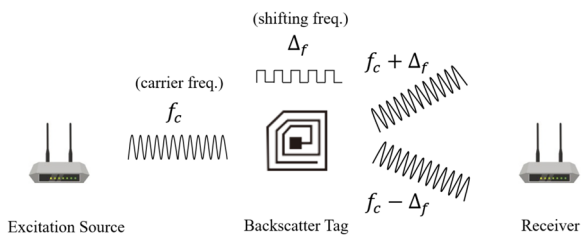


Fig. 2 Illustration of backscatter communications. The backscatter tag reflects the carrier signal from the excitation source and conveys its data to the receiver

$$P_r = P_b \frac{\left|\Gamma_1^* - \Gamma_2^*\right|^2}{4} \tag{1}$$

where P_b is the received power at the tag from the excitation signal, Γ_i^* represents the complex conjugates of the reflecting coefficient, and $\Gamma = \frac{Z_a - Z_c}{Z_a + Z_c}$, with Z_a being the impedance of antenna and Z_c the impedance of RF circuit. In order to improve the power efficiency of the backscatter link as much as possible, the tag *shall* maximize the reflecting coefficients of two impedance states. Hence, the impedance of RF circuit at the tag is controlled by connecting and disconnecting the antenna to the ground through an RF switch.

In order to further eliminate the interference from the legacy link (conventional communication links), the tag generates a square wave with frequency Δ_f, and controls the RF switch. According to the basic principle of the Fourier transform, a square wave is a linear combination of sinusoidal signals with harmonic frequencies. Thereby, the square wave contains a harmonic component as a sinusoidal $\sin(2\pi\Delta_f t)$. The backscatter signal is then described as

$$\sin(2\pi f_c t)\sin(2\pi\Delta_f t) = (\cos(2\pi(f_c - \Delta_f)t) - \cos(2\pi(f_c + \Delta_f)t))/2 \tag{2}$$

with a frequency shift Δ_f.

Instead of using a dedicated excitation source, WiFi backscatter systems further utilize the existing WiFi infrastructure and modulate the data on the ambient WiFi signals. It can then be significantly ubiquitous and with low deployment cost.

System design

Problem domain

NuWa deals with the problem of how to enable WiFi backscatter communication with the WiFi traffic having random on-off states.

We evaluate the performance of WiFi backscattering in a typical real-life environment, where WiFi traffic is not dedicated nor controlled to facilitate backscatter communication. In particular,

- NuWa tackles scenarios where there is no chance to inject dummy WiFi packets to the excitation channel, *i.e.*, the WiFi traffic is not manipulated.
- NuWa applies to scenarios when the duration of the off-states does not exceed the maximum error-correcting capability of the designed codes.
- NuWa applies to scenarios where the preamble needs to be correctly delivered to the receiver. If the preamble is corrupted, it is impossible to efficiently detect the frame at the receiver.

System overview

The functionality of the designed NuWa is divided into the tag and the receiver portions, as shown in Fig. 3.

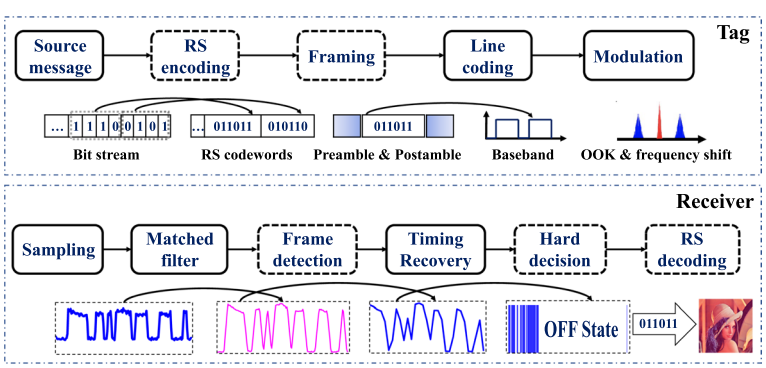


Fig. 3 The system overview of NuWa. It adopts look-up table-implemented RS code to protect the backscattering data. We aim to optimize the design and implementation of RS codes. Furthermore, we design a new frame structure to improve the frame detection in the proposed system

At the tag, the source message is encoded by the RS encoder using the designed lookup table. The framing process is then followed, which inserts the preamble before and appends the postamble after the coded sequence. The formed frame is further converted to a baseband signal using a line code, such as non-return zero. Finally, the baseband signal is modulated using on-off keying through a radio-frequency switch.

At the receiver, a series of regular receiving operations in communication systems are implemented. If a packet is detected, the receiver starts to take the samples of the signal, and performs a matched filter to improve the signal quality. It then detects the frame and extracts the payload from the right positions, according to the timing recovery. Finally, we make a hard decision and decode the application layer message using an RS decoder.

Transmission scheme

RS encoding process

The source message is encoded using RS code (refer to Appendix for the detail of RS encoding and decoding with an example) and is inserted into the payload.

To perform RS encoding, every $m = \log_2(n+1)$ bits of the source message are grouped together and converted to one symbol in Galois field GF (2^m). Then NuWa divides the symbols into k blocks and computes the $(n-k)$ parity symbol for each block, of which the coding rate is $\frac{k}{n}$. The parity symbols are finally padded after the k message symbols to form an RS(n, k) code. According to the coding theory, an RS(n, k) code is able to correct $\lfloor \frac{n-k}{2} \rfloor$-symbol errors, i.e., $m\lfloor \frac{n-k}{2} \rfloor$-bit errors. For instance, if we choose $m = 6$ and use $\frac{11}{63}$ coding rate, it can correct up to 156-bit burst errors, which corresponds to a 156μs interval if the backscattering rate is 1 Mbps. Therefore, RS code can effectively combat the off states in the excitation signal.

Choosing different m generates different lengths of payload data and thus affects the probability of encountering off states (the longer the length, the higher probability). In our design, we choose small m values: $m = 5$ for 32-byte payload and $m = 6$ for 64-byte payload. The reasons are threefold: (1) In typical applications of backscatter communication, the source message is usually short. Therefore, it is not superior to choose a large m where a lot of 0's need to be padded to the source message to generate RS codeword in the Galois field. (2) A long frame may span several off states and have more burst errors. Hence, it requires a lower rate code for error correction which is inefficient. (3) A long frame requires more memory space to temporarily hold the frame data for processing. However, the storage size at a low-price source-restricted tag is quite limited. However, the more important thing is the coding rate which not only determines the burst error correcting capability but also affects the goodput performance (the useful data delivered per unit of time). Intuitively, if the excitation traffic is dense with short intervals, we should use a high coding rate because only a short length of data is lost. Rather, if the traffic is sparse, a lower coding rate achieves better error correction performance. Hence, the coding rate highly depends on the "off" duration of the excitation signal.

Input: Randomly generated packet duration and interval in μs.
Output: n^*, k^*

```
fer = 1;
n_0 = 1;
k_0 = 0;
for m ← 3; m ≤ 7; m++ do
    ñ ← 2^m − 1;
    for k ← ñ − 2; k ≥ 3; k = k − 2 do
        FER ← result evaluated by simulations;
        if (1 − FER)k/n > (1 − fer)k_0/n_0 then
            k* ← k;
            n* ← ñ;
            fer ← FER; n_0 ← ñ; k_0 ← k;
        else
            break;
        end if
    end for
end for
```

Algorithm 1 Bruteforce search of the optimal RS code.

Traffic-based Coding rate selection. To determine how many parity symbols are required, we use a heuristic search method to find the proper coding rate, which is summarized in Algorithm 1. Based on the duration of the off-state intervals, we consider dividing the random excitation into three typical cases: high, medium, and low traffic. We then ran simulations with collected real-life traffic traces to determine the required coding rate for backscattering different sizes of packets. The selection policy on the coding rate is determined by the gain on goodput performance. For example, we want to choose a suitable code from RS(n_1, k_1) and RS(n_2, k_2), if

$$\frac{1 - FER_1}{1 - FER_2} > \frac{k_2/n_2}{k_1/n_1}, \tag{3}$$

where FER_1 and FER_2 are the frame-error-rate respectively achieved by RS(n_1, k_1) and RS(n_2, k_2), then we select RS(n_1, k_1) code.

The reason is that RS(n_1, k_1) has better goodput performance. Algorithm 1 is executed using the selection policy to choose RS code in a limited subset of the RS codes. In fact, the tag cannot perform RS coding if n is too large due to the memory limitation as aforementioned. While in the practical deployment, the coding

rate *shall* dynamically adjust according to the real-time traffic patterns, which is left as a future study.

Real-time consideration of the encoding. The encoding of RS code is performed in the Galois field, which usually requires a lot of computational resources to generate the parity symbols [24]. Unfortunately, the power and computational resources are scarce at the low-cost tag (MSP432 platform). The problem is then how can the tag still perform RS encoding in real-time since the tag needs to send data at any time.

We built an RS library *mspRS* in C based on [25] using lookup tables (LUT). Since we use Galois field with characteristic 2 [26], the addition and subtraction are identical to the XOR operation and can be efficiently done in C. For multiplication and division, the LUT is adopted to speed up these operations.

In order to further optimize the encoding process at MSP432 platform, we improve the encoding process by re-designing the LUT. First, the group of codes with the same value m shares the common tables for logarithmic and inverse-logarithmic operations. Second, we also pre-compute the generator polynomial table.

Framing process

In wireless communication systems, a preamble is typically inserted before each packet for the receiver to detect whether a packet is present or not. Similarly, NuWa also adopts a preamble before each frame to indicate the tag is backscattering a frame. As shown in Fig. 4, the problem here is that the preamble may encounter a short off-state in the WiFi traffic. If it happens, the whole packet is dropped. However, the payload of the packet may not be affected by the off state, or it is possible to recover by the decoding process. It is thus wasting resources if the whole packet is dropped for backscatter communications. Hence, we need to carefully design the frame structure to combat the preamble loss.

Part of the long preamble, *e.g.*, lasting 144 μs in 802.11b protocol, can easily be corrupted by the off-state. On the other hand, if we use a short preamble, the entire preamble may fall into the off-state interval.

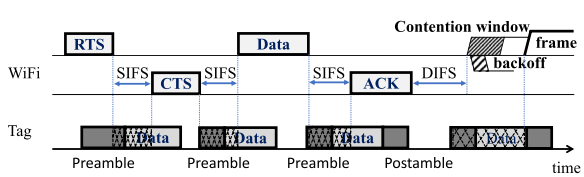


Fig. 4 The preamble design of the tag. A longer preamble has a higher probability of encountering the "on" state. While a shorter preamble may drop into the "off" state. NuWa adopts both preamble and postamble to increase the opportunities of detecting the incoming packets

The best way to protect the preamble is that we distribute multiple preambles within a frame, but this brings in a lot of overheads and increases the complexity of packet detection. To this end, we choose Barker codes and add them not only before the packet (preamble), but also after the packet (*postamble*). By doing so, the short preamble is repeated twice and is separated by the payload data. The possibility of both preamble and postamble being corrupted is smaller compared with one single preamble. Furthermore, in order to easily determine whether the detected Barker code is preamble or postamble, we flip the Barker code so the postamble is not repeating but symmetrical to the preamble.

In summary, as shown in Fig. 5, each frame consists of the following fields.

- Preamble: for frame detection and adaptive threshold calculation, Barker codes with lengths 7, 11, 13 bits are adopted respectively.
- 2-byte header: includes start frame delimiter (SFD) and the length of the payload.
- Payload data: the RS codeword of data.
- 2-bytes cyclic redundancy check: detecting any in-transit corruption of data.
- Postamble: the reversed version of the preamble for packet detection, which can also be easily distinguished from the preamble.

The frame is then modulated by a line code, in particular, a non-return zero (NRZ) code. Finally, the sequence is backscattered by on-off keying modulation (OOK).

Receiver structure

As shown in Fig. 3, the receiver listens to the shifted frequency and starts its processing once a frame is detected. We now explain each functional block of the receiver in decoding the message.

Frame detection

Sampling and Matched filter. The receiver first executes a sampling process to take I/Q samples and calculates the amplitude of the received signal. A matched filter is then adopted for improving the signal-to-noise ratio of the received signal. Since the baseband signal is a square wave. we take the amplitude of the received signal, then

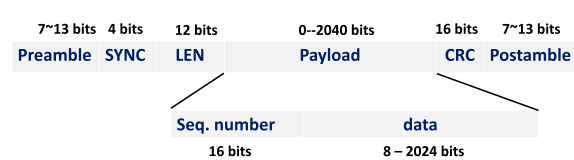


Fig. 5 Frame definition of NuWa. We append a postamble after the payload to increase the opportunity of frame detection

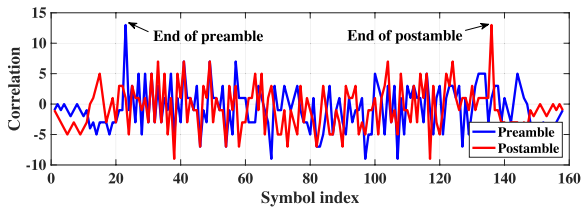


Fig. 6 Find the starting or the ending positions using preamble and postamble detection through the correlation values

perform a convolution between the amplitude of the signal and a square wave. The output is fed to the forward and backward detection block to detect whether the tag is backscattering.

Forward and backward detection. We adopt both preamble and postamble in the frame for improving the robustness of frame detection against the off-state intervals of the excitation signal. For detecting the preamble and the postamble, we calculate the auto-correlation using a sliding window. The peak value appears at the end of the preamble with a forward window, or the start of the postamble with a backward window. As demonstrated in Fig. 6, in the forward phase, once a peak value is detected, the receiver starts buffering the samples. Otherwise, in the backward phase, if the peak value is detected, the receiver takes the buffered samples in the cache from the back. The samples in the cache are dropped if neither the preamble nor the postamble is detected.

Timing recovery and downsampling

After that, a downsampling process is adopted to filter the samples. However, the timing synchronization between the tag and the receiver should be properly established. Otherwise, the error probability significantly increases with inferior timing recovery. At the receiver, the symbol synchronizer based on an interpolation filter, zero-crossing timing error detector, and modular-1 interpolation control are implemented before the downsampling process. After synchronizing, we take samples with the baseband frequency according to the timing recovery. The down-sampled symbols are then converted to a bit stream by comparing the amplitude value with an adaptive threshold which is described in detail below.

Hard decision with adaptive thresholds

NuWa performs the following process to obtain the detection threshold. A cross-correlation process is performed between the known preamble pattern and the received samples. When the received preamble is correlated with the known preamble, a peak will be generated. Once the peak is detected among the cross-correlation values, of which the position is considered as the end point of the preamble. The threshold is then set as the average value of the minimal amplitude representing "1" and the maximal amplitude representing "0" in the samples. It should be mentioned here that the advantage of performing adaptive threshold adjustment is that the received signal power strength varies considerably in the context of backscatter communications.

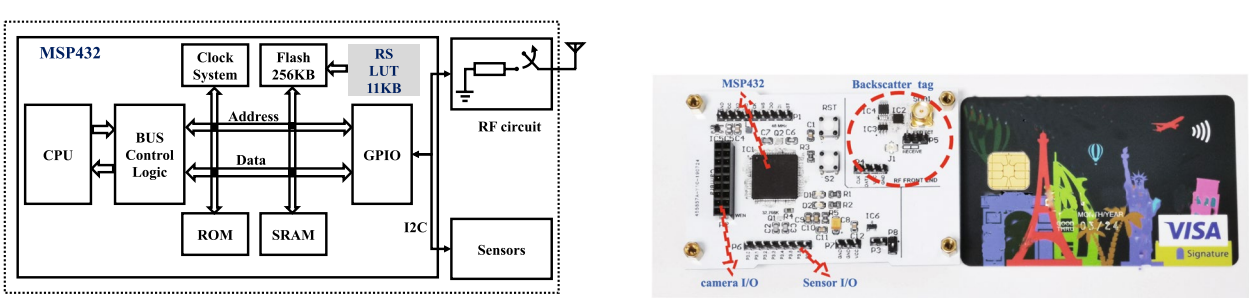
RS decoding process

The bit stream is finally forwarded to the decoder using implemented Berlekamp-Massey decoding algorithm to decode the source symbols. After decoding is completed, the symbols are converted back to a bit stream. Finally, the bit stream is fed into the source decoder which maps the bits to the understandable form, *e.g.*, image.

Implementation of NuWa

We implemented a prototype of our system using a commodity WiFi transmitter, a self-designed backscatter tag and a USRP-based receiver. Our equipment is shown in Fig. 7. We describe their implementation details below.

Receiver implementation: Our receiver is implemented on the USRP B205 platform with the GNU Radio software.



(a) Hardware architecture of NuWa. (b) Hardware prototype.

Fig. 7 The implementation of hardware prototype. We use the MSP432 microcontroller to process the data and control the backscatter tag for backscattering. **a** Control flow and data flow in our design. **b** Hardware prototype

Tag implementation: We designed and implemented our tag based on the low-power Texas Instrumental (TI) MSP432 platform. The tag includes three main components: 1) A MSP432P401R micro-controller chip [27] and its peripheral circuits; 2) GPIO ports for connecting sensors and 3) An RF front-end implemented by an analog switch (ADG902) for turning on/off the antenna and shifting the carrier frequency. The ADG902 is then controlled by another RF switch (ADG918) for modulating the digital signal sent by the MSP432 microcontroller. All components are placed on a single 2-layer Printed Circuit Board (PCB) with a size of 8.4 $cm \times 5.6\,cm$. The thickness is 0.8 mm. The current size is similar to a credit card and the size could be further reduced[4].

Evaluation and results

We evaluate the performance of NuWa in different environments in terms of the following metrics: BER, FER, throughput, and goodput. To make a comparison with the best case, we assume an ideal *omniscient* scheme which exactly knows the WiFi on-off states. Note that this is not practical in reality with roadside APs and we just employ this scheme for comparison purposes. The results obtained from the experiments using our prototype verify the following.

- NuWa achieves a very close performance compared with the ideal omniscient scheme.
- Our system achieves a 1 Mbps throughput (600 kbps goodput) when the tag is located 1 m away from the receiver and 0.5 m away from the WiFi AP.
- Our system achieves a throughput 950 kbps if the tag is 6 m away from the receiver.

Experimental setup

We randomly generate source data and backscatter them frame-by-frame to evaluate the average BER, FER, throughput and goodput. For each setting, the tag backscatters 10,000 frames in total. We use a USRP B205 to receive the tag's data at a sampling frequency of 2 MHz for a 400 kbps data rate and 5 MHz for a 1 Mbps data rate.

We conduct experiments in a lab with a size of $9 \times 8\text{m}^2$, of which the floor plan is shown in Fig. 8. The backscatter tag is located 0.5 m and 1 m away from the WiFi AP and the receiver, respectively by default. We use Channel 1 with a central frequency 2.412 GHz and 20 MHz bandwidth for WiFi AP. *Please be noted that we use a USRP to work as the WiFi AP in order to focus on our main issue, i.e., the intermittent excitation.* The tag shifts the incoming excitation signal by 24 MHz, i.e., the signal is shifted to frequency 2.436 GHz.

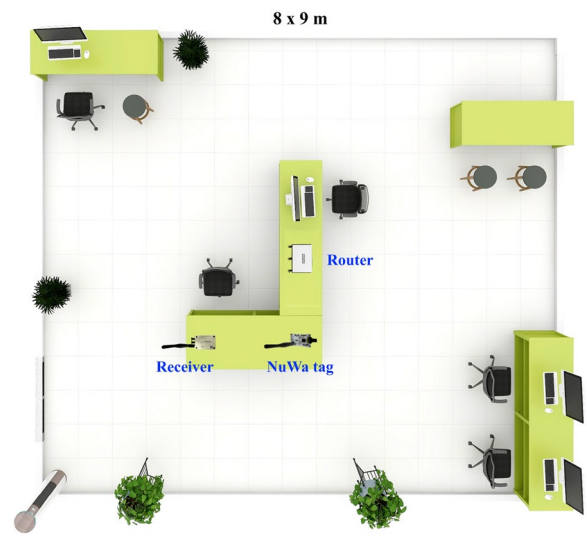


Fig. 8 The floor plan of the office where the experiments are conducted

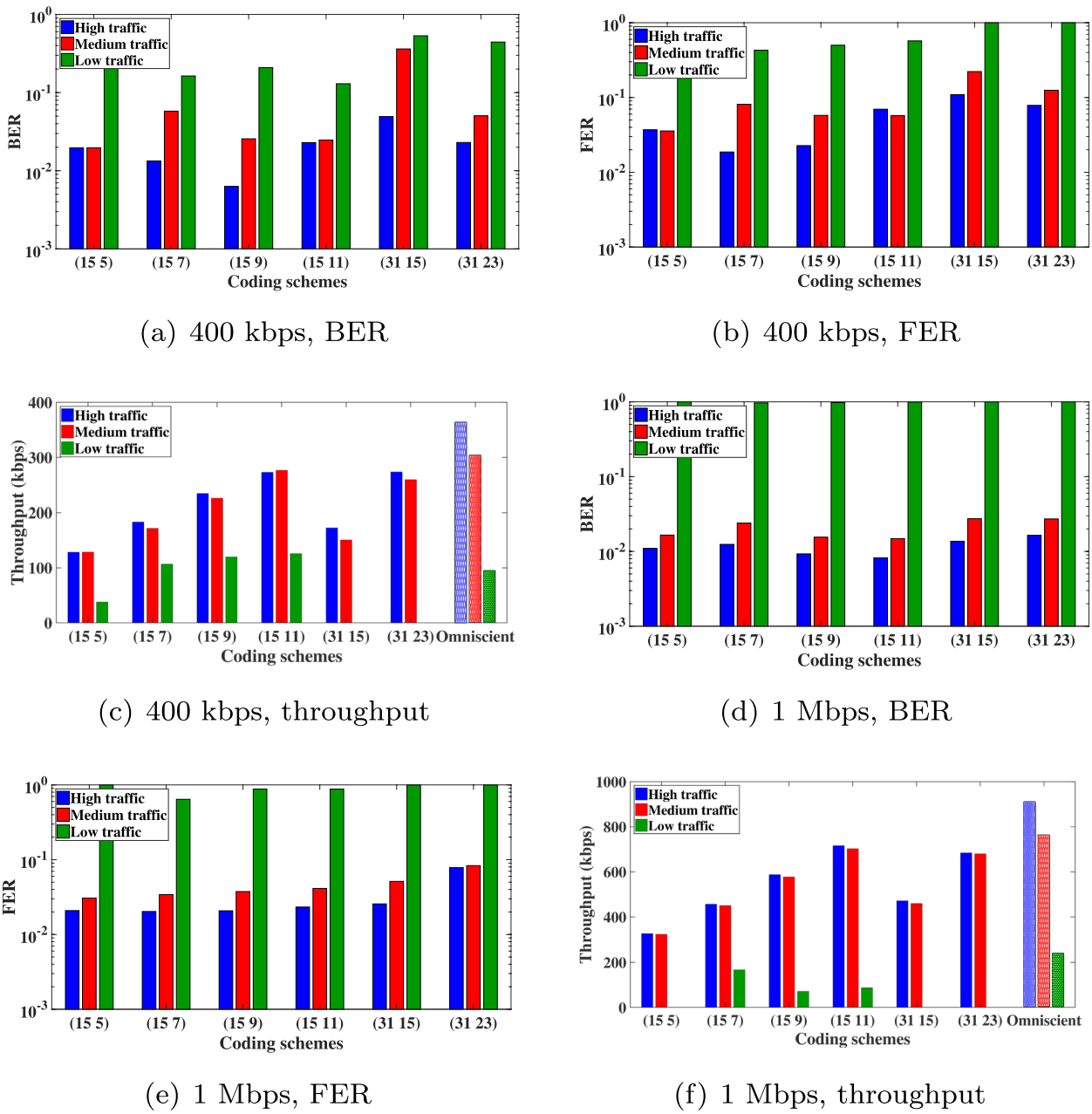
To simplify the evaluation process, we generally consider three different cases of the excitation signals, including (1) high traffic: long frame duration with short interval; (2) medium traffic: both frame duration and interval are moderate and (3) low traffic: short frame duration but long interval. In particular, the excitation source (frame) duration is fixed in the range [70, 8800] μs (The range is obtained from the captured large amount of WiFi packets.). Moreover, three different interval distributions are adopted to distinguish the traffic conditions. We set the uniform distribution of interval length to [28, 1000], [28, 5000] and [28, 10000] μs for high, medium and low traffics, respectively.

Experimental results

Impact of coding rates. We now evaluate the system performance when the tag adopts different coding rates. The WiFi AP-tag distance is set as 0.5 m and the tag-RX distance is set as 1 m. We generate WiFi excitation signals, of which the duration and interval follow different uniform distributions respectively. The tag uses different coding rates to encode its message.

As we can see from Fig. 9(a)-(c), the performance varies when adopting different coding rates. In general, a low coding rate leads to low FER. However, we want to point out that it is not a good choice of using a low coding rate if the FER performance can only be slightly improved.

[4] Reducing the size of the hardware is not the goal of this work and we leave it as our future study since our primary goal is to prove that using a well-designed code can improve the performance of backscatter systems.



(a) 400 kbps, BER

(b) 400 kbps, FER

(c) 400 kbps, throughput

(d) 1 Mbps, BER

(e) 1 Mbps, FER

(f) 1 Mbps, throughput

Fig. 9 The BER, FER, and throughput performance comparison between NuWa and baseline backscatter system, in different traffic conditions

For example, in Fig. 9(e), the FER is almost the same for high-traffic cases (blue color). In such cases, a low coding rate has a much higher overhead but the performance is not improved. To more clearly understand this phenomenon, we calculate the goodput performance of NuWa, and plot in Fig. 10. It is clear that NuWa with RS(31, 23) achieves the best goodput performance. Therefore, it is of great importance to choose a proper coding rate for different traffic patterns.

Impacts of frame intervals. We then evaluate the impact of frame intervals between the WiFi frames on system performance, of which the results are shown in Fig. 9(a)-(f). From the results, it is found that NuWa is able to fight against the excitation off state and backscatters its message. Also, NuWa in high traffic loading case outperforms the other cases. When the interval is relatively long which corresponds to the low traffic case, our system still achieves around 150 kbps throughput.

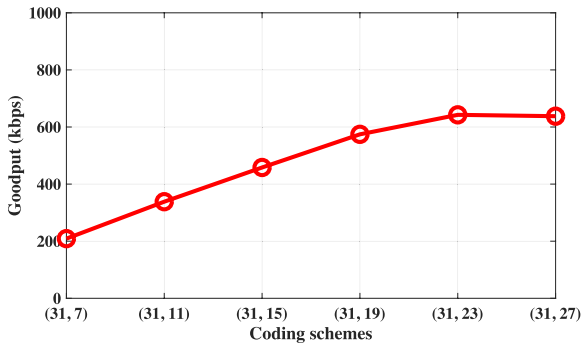


Fig. 10 The goodput performance of each encoding scheme

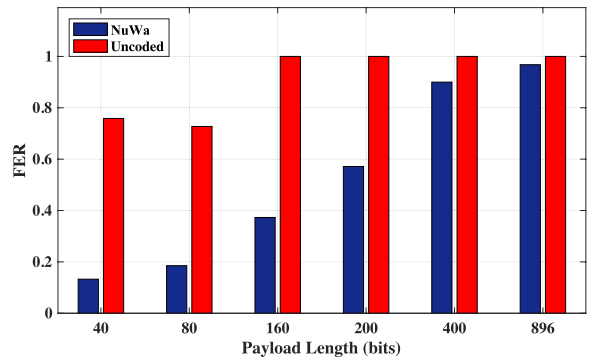


Fig. 12 The impact of payload length on the frame error rate

Impacts of different types of preambles. We evaluate the performance of our preamble design. We adopt three types of preambles (Barker code with a length of 7, 11, and 13 bits respectively) to evaluate the number of detected frames within a unit of time.

The receiver would not start buffering samples if it does not detect the preamble or postamble correctly. With a longer preamble, the chance of the preamble being corrupted may increase. We thus aim to examine the impact of the preamble length on system performance. We detected 833, 1436, and 576 frames out of 2000 backscattering frames with the 3 different lengths of preambles. We see that with a length of 11 bits, the detection rate is the highest among the three. The reason is that, if the preamble is too short, the correlation peak may not be high enough and can be interfered with noise. While a long preamble has a higher probability of falling in the off-state intervals. Thus a trade-off can be found here. In the later experiment, we use Barker code with a length of 11 bits as the preamble and postamble.

Impacts of the frame design. Furthermore, we plot the frame detection percentage by preamble and postamble in Fig. 11. Based on the results, we can observe that using postamble does improve the frame detection rate by more than 10%.

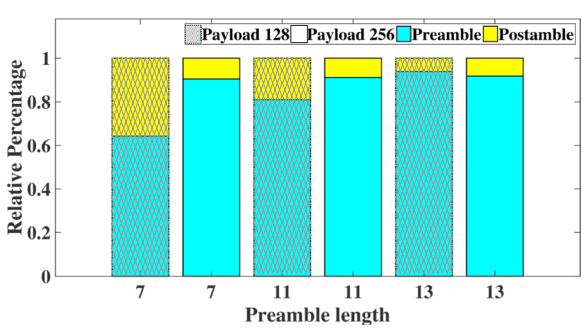


Fig. 11 The impact of preamble length on the frame detection

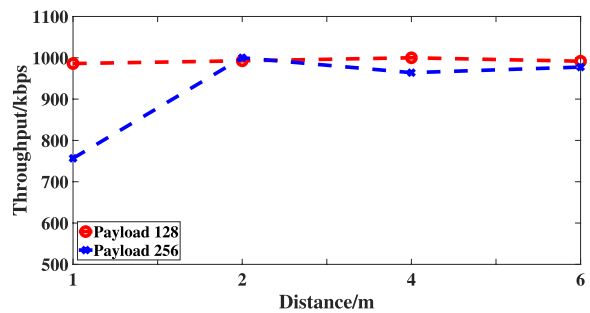


Fig. 13 The impact of the distance between the tag and the receiver

We now evaluate the impact of the payload length and backscatter rate. Intuitively, the probability of encountering an "off" state is higher if the frame lasts longer. In other words, we should use stronger RS code to protect the backscatter communication if a longer frame is adopted. We fix the backscatter rate at 400 Kbps and vary the payload length. The obtained FER results are plotted in Fig. 12. We observe that as the frame becomes longer, FER increases gradually. For comparison, we also plot the FER of the uncoded case as a baseline[5]. It is found that our system outperforms the baseline system and the baseline system will drop almost all the frames if the frame is larger than 160 bits.

Different distances between Tag-RX. Figure 13 illustrates the throughput performance, with different Tag-RX distances, and the distance between the tag and the router is set at 0.5 m. We use the coding rate obtained by Algorithm 1 in the high traffic loading condition, which is RS(15, 7) for 128 bit payload and RS(31, 15) for 256 bit payload. From the results, it is found that increasing distance slightly affects the throughput of NuWa, *i.e.*, NuWa achieves very stable throughput if the tag-RX distance is within several meters. Among the results, the best

[5] In this paper "uncoded" means without RS encoding.

throughput is 1 Mbps if the tag-RX distance is 2 *m*. Our system could still achieve around 950 kbps if the distance is increased to 6 *m*.

Discussion

Potential applications

In this work, we implemented a backscatter tag in the MSP432 microcontroller platform, which can connect sensors using *I2C* protocol. NuWa mainly applies to the scenarios requiring around a hundred kbps in transmission rate but the power supply is difficult. On one hand, we can connect a camera sensor in the buildings to monitor the security or the water meter if the power cable and maintenance are difficult. On the other hand, the platform can be deployed to evaluate air quality by connecting appropriate sensors.

Limitations

In this work, we adopt a USRP as the receiver, which is a dedicated device with a high cost. However, the primal goal of designing NuWa is to prove the possibility of fighting against the random characteristics of the WiFi signals. We hence design the Reed-Solomon code in the application layer to protect the message, and the frame structure in the link layer to improve the system performance. In fact, our system can be implemented on top of the proposed code translation technique [12] to enable the decoding at the off-the-shelf WiFi transceivers. This is left as a future study.

Another limitation of NuWa is that we do not collect WiFi data to form a WiFi packet dataset for optimization of the RS codes. Instead, we just use a random distribution to generate the excitation. Additionally, the energy of each packet is not taken into account. To further expand this work, we can generate a fine-grained energy map for the optimization of the WiFi backscatter.

Finally, the receiving function of the tag is not debugged adequately for the feedback channel of the backscatter link.

Conclusion

WiFi backscatter communication suffers from low reliability due to the on-off variation nature of the WiFi traffic. NuWa aims to tackle the backscattering outage during off-states. We proposed a lookup table-based RS code design and a brute-force search-based RS coding rate selection algorithm. Furthermore, we jointly designed the RS encoding process and the frame structure which flexibly fit different traffic loading conditions. Finally, we implemented NuWa using MSP432. It was found from the results that NuWa can tolerate off state in the excitation signals. NuWa achieved a 1 Mbps throughput when the tag is located 1 *m* away from the receiver, and 950 kbps if the distance is 6 *m*. Moreover, when the traffic loading is very light, NuWa still achieved a 150 kbps throughput. Hence, NuWa is highly reliable even if the excitation signal has randomly alternated on and off states. We plan to implement NuWa by using COTS WiFi routers and propose a protocol stack in the future study.

Appendix

Reed-Solomon (RS) code

As with any error-correcting code, the principle of an RS code is to transform a given information sequence into a longer sequence called a codeword by adding some redundancy. In this way, the information can be recovered by performing a decoding process. A Reed-Solomon code is specified as RS(n, k) with m-bit symbols. The RS encoder takes k data symbols of m bits each and adds parity symbols to construct a n symbol codeword. There are $n-k$ parity symbols in a codeword for protecting the data symbols. RS codes are defined over symbols, i.e., over a Galois field (GF) of size 2^m. An RS code can correct $t = \lfloor \frac{n-k}{2} \rfloor$ symbol errors, and are optimal in this regard — RS codes achieve the Singleton bound with equality. As shown in Fig. 14, for GF(8), each symbol is represented by 3 bits. If one or more bits have an error, then the whole symbol is in error. Since the RS code corrects symbol errors, it does not matter if one or all 3 bits have an error – it only counts as 1 out of t symbols to be corrected. However, a 3-bit burst error could span up to two symbols, and in this case, the RS code must correct 2 out of t symbols. This is the reason RS codes work well for burst-error correction.

RS encoding. Let $\mathbf{\Phi}(n \times k)$ be a generator matrix of an RS(n, k) code. If $\mathbf{u}$ is a $k \times 1$ vector of information symbols from GF(2^m) (consisting of $k \cdot m$ bits), then $\mathbf{c} = \mathbf{\Phi} \cdot \mathbf{u}$ is an $n \times 1$ codeword of symbols from GF(2^m) (consisting of $n \cdot m$ bits). A matrix $\mathbf{\Phi}$ such that any k rows linearly independent exist, and we are interested in the

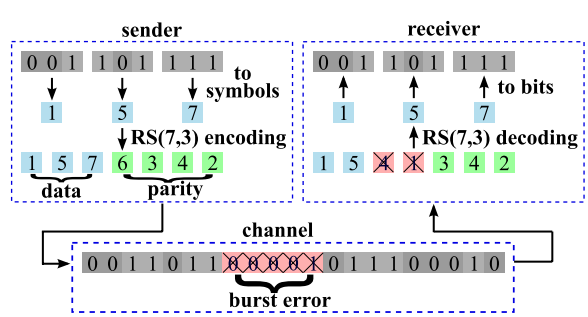


Fig. 14 Example of RS encoding and decoding

case that $\mathbf{\Phi}$ generates a systematic code. That is, k input data words are explicitly present among the n generated codewords. That is, $\mathbf{\Phi}_{\text{sys}} = [\mathbf{I}_k | \mathbf{\Psi}_{k,n-k}]$ and the information sequence $\mathbf{u}$ appears in the first k positions of the codeword $\mathbf{c}$.

RS encoding complexity. We are primarily concerned with the encoding complexity, which takes place at the tag. While the Vandermonde form for the RS generator matrix is well known, recently it has been shown that the Hankel matrix has lower complexity when encoding systematic RS codes [24]. This approach may be suitable when the RS code parameters are dynamic. Alternatively, the parity portion of the generator matrix $\mathbf{\Psi}$ may be generated once and stored. The storage requirement is $k(n-k)$, and the computational requirement is also proportional to $k(n-k)$, with mathematical operations over the GF(2^m).

RS decoding. If t or fewer symbol errors occur during transmission, then an RS decoder can recover the transmitted RS codeword and the transmitted data is in positions 1 to k of this codeword. The design of efficient decoding algorithms has been the study of intense research for years, and now both hardware and software implementations are widely used in practice. The best known of these is the Berlekamp-Massey decoding algorithm, which finds the roots of an error-locator polynomial. While the complexity of the most efficient algorithms is proportional to $n \log n$, this operation is performed at the receiver and does not contribute to the tag's computational burden.

Acknowledgements
The authors would like to thank anonymous reviewers for their valuable comments on the manuscript.

Authors' contributions
Zhiyi Yang and Guiping Lin prepared the manuscript. Weiwei Jiang conducted the experiments. Xin He, Weiwei Jiang, Yong Xu, and Panlong Yang discussed the main idea and the experimental protocol of this paper. Yujun Zhu revised the manuscript. Xin He is the supervisor of Zhiyi Yang and Guiping Lin.

Funding
This work has been in part supported by the Natural Science Foundation of China under grants No. 62072004, 61702011.

Availability of data and materials
The authors will provide the data and materials on demand.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 2 December 2022 Accepted: 14 August 2023
Published online: 16 November 2023

References

1. Talla V, Kellogg B, Ransford B, Naderiparizi S, Gollakota S, Smith JR (2015) Powering the next billion devices with Wi-Fi. In: Proc. of ACM CoNEXT, ACM, New York
2. Liu V, Parks A, Talla V, Gollakota S, Wetherall D, Smith JR (2013) Ambient backscatter: Wireless communication out of thin air. In: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, ACM, New York, SIGCOMM '13, pp 39–50
3. Sample AP, Yeager DJ, Powledge PS, Mamishev AV, Smith JR (2008) Design of an rfid-based battery-free programmable sensing platform. IEEE Trans Instrum Meas 57(11):2608–2615. https://doi.org/10.1109/TIM.2008.925019
4. Wang A, Iyer V, Talla V, Smith JR, Gollakota S (2017) FM backscatter: Enabling connected cities and smart fabrics. In: Proc. of USENIX NSDI, USENIX Association, Boston, pp 243–258
5. Kellogg B, Parks A, Gollakota S, Smith JR, Wetherall D (2014) Wi-Fi backscatter: Internet connectivity for RF-powered devices. In: Proc. of ACM SIGCOMM, ACM, New York, pp 607–618. https://doi.org/10.1145/2619239.2626319
6. Bharadia D, Joshi KR, Kotaru M, Katti S (2015) BackFi: High throughput wifi backscatter. In: Proc. of ACM SIGCOMM, ACM, New York, pp 283–296. https://doi.org/10.1145/2785956.2787490
7. Li P, He X, Freris NM, Yang P (2020) Capacity analysis of ambient backscatter system with bernoulli distributed excitation. In: Yu D, Dressler F, Yu J (eds) Wireless Algorithms, Systems, and Applications. Springer International Publishing, Cham, pp 218–230
8. Huang J, Xing G, Zhou G, Zhou R (2010) Beyond co-existence: Exploiting WiFi white space for Zigbee performance assurance. In: Proc. of IEEE ICNP, IEEE, Kyoto, pp 305–314
9. Yan Y, Yang P, Li X, Tao Y, Zhang L, You L (2013) ZIMO: Building cross-technology MIMO to harmonize zigbee smog with wifi flash without intervention. In: Proc. of ACM MobiCom, ACM, Miami, pp 465–476. https://doi.org/10.1145/2500423.2500426
10. Zhang P, Josephson C, Bharadia D, Katti S (2017) FreeRider: Backscatter communication using commodity radios. In: Proc. of CoNEXT, ACM, New York
11. Kellogg B, Talla V, Smith JR, Gollakot S (2016) PASSIVE WI-FI: Bringing low power to Wi-Fi transmissions. In: Proc. of USENIX NSDI
12. Zhang P, Bharadia D, Joshi K, Katti S (2016) HitchHike: Practical backscatter using commodity wifi. In: Proc. of ACM SenSys, ACM, New York
13. Kim T, Lee W (2018) Exploiting residual channel for implicit Wi-Fi backscatter networks. In: Proc. of IEEE INFOCOM, pp 1–9
14. Zhang P, Bharadia D, Joshi K, Katti S (2016) Enabling backscatter communication among commodity wifi radios. In: Proc. of ACM SIGCOMM, ACM, Florianópolis, SIGCOMM '16, pp 611–612. https://doi.org/10.1145/2934872.2959072
15. Zhao J, Gong W, Liu J (2018) X-Tandem: Towards multi-hop backscatter communication with commodity WiFi. In: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, ACM, New York, MobiCom '18, pp 497–511. https://doi.org/10.1145/3241539.3241553
16. Mi N, Zhang X, He X, Xiong J, Xiao M, Li XY, Yang P (2019) CBMA: Coded-backscatter multiple access. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp 799–809. https://doi.org/10.1109/ICDCS.2019.00084
17. Talla V, Hessar M, Kellogg B, Najafi A, Smith JR, Gollakota S (2017) Lora backscatter: Enabling the vision of ubiquitous connectivity. Proc of ACM Ubicomp 1(3):105:1–105:24. https://doi.org/10.1145/3130970
18. Peng Y, Shangguan L, Hu Y, Qian Y, Lin X, Chen X, Fang D, Jamieson K (2018) PLoRa: A passive long-range data network from ambient loRa transmissions. In: Proc. of ACM SIGCOMM, ACM, New York, pp 147–160. https://doi.org/10.1145/3230543.3230567
19. Yang C, Gummeson J, Sample A (2017) Riding the airways: Ultra-wideband ambient backscatter via commercial broadcast systems. In: Proc. of IEEE INFOCOM, pp 1–9
20. He X, Jiang W, Cheng M, Zhou X, Yang P, Kurkoski B (2020) Guardrider: Reliable WiFi backscatter using reed-Solomon codes with QoS guarantee. In: 2020 IEEE/ACM 28th IWQoS, IEEE, Hangzhou, pp 1–10. https://doi.org/10.1109/IWQoS49365.2020.9213057
21. Wang Q, Chen S, Zhao J, Wei G (2021) Rapidrider: Efficient wifi backscatter with uncontrolled ambient signals. In: Proc. of IEEE INFOCOM, IEEE, Virtual Conference, pp 1–10

22. Xu S, He X, Wu F, Lin G, Yang P (2022) Design on rateless LDPC codes for reliable Wifi backscatter communications. In: Wang L, Segal M, Chen J, Qiu T (eds) Wireless Algorithms, Systems, and Applications. Springer Nature Switzerland, Cham, pp 59–71
23. (2012) IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Std 80211-2012 (Revision of IEEE Std 80211-2007) pp 1–2793
24. Mattoussi F, Roca V, Sayadi B (2012) Complexity comparison of the use of vandermonde versus hankel matrices to build systematic mds reed-solomon codes. In: Proc. of SPAWC
25. Python Library for Reed-Solomon Code. https://github.com/tomerfiliba/reedsolomon. Accessed 20 Dec 2022
26. Peterson WW, Peterson W, Weldon E, Weldon E (1972) Error-correcting codes. MIT Press
27. MSP432P401R datasheet. https://www.ti.com/lit/ds/symlink/msp432p401r.pdf. Accessed 17 Nov 2022

Publisher's Note