Spatial Stream Backscatter Using Commodity WiFi

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

Wei Gong
University of Science and Technology of China, China
Simon Fraser University, Canada
weigong@ustc.edu.cn

Jiangchuan Liu
Simon Fraser University
Burnaby, Canada
jcliu@cs.sfu.ca

ABSTRACT

Backscatter WiFi offers a novel low-cost and low-energy solution for RFID tags to communicate with existing WiFi devices. State-of-the-art backscatter WiFi solutions have seldom explored advanced features in the latest WiFi standards, in particular, *spatial multiplexing*, which has been the cornerstone for 802.11n and beyond. In this paper, we present MOXcatter, a WiFi backscatter communication system that works with spatial streams using commodity radios, while keeping the ongoing data communication unaffected. In MOXcatter, a backscatter tag can embed its sensing data on ambient spatial-stream packets, and both the sensing data and the original packets can be decoded by commodity WiFi devices. We have built a MOXcatter prototype with FPGAs and commodity WiFi devices. The experiments show that MOXcatter achieves up to 50 Kbps throughput for a single stream and up to 1 Kbps for double streams with a communication range (tag-to-RX) up to 14 m. We discuss the tradeoffs therein and possible enhancements, and also showcase the applicability of our design through a sensor communication system.

CCS CONCEPTS

• **Networks** → **Cyber-physical networks**; **Network architectures**; **Sensor networks**;

KEYWORDS

WiFi Backscatter, Spatial Stream, Internet of Things

ACM Reference Format:
Jia Zhao, Wei Gong, and Jiangchuan Liu. 2018. Spatial Stream Backscatter Using Commodity WiFi. In *Proceedings of MobiSys'18, Munich, Germany, June 10-15, 2018.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3210240.3210329

1 INTRODUCTION

As the most widely used indoor wireless technology, 802.11 WiFi has evolved to become a prominent choice of Internet of Things (IoT) applications. For example, Google Home, a popular smarthome control product, uses WiFi to stream media content directly from/to the cloud, so for integrating such ambient sensors/controllers as the NEST thermometers, smoke alarms, and cams. Yet the energy consumption of WiFi has long been a concern: A typical WiFi transceiver consumes 80 mW power to send 75 bytes data per second, which is 40x higher than Bluetooth does [1].

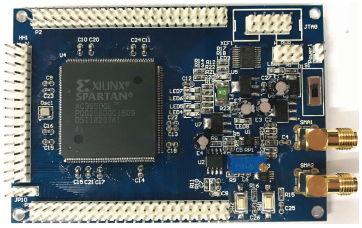


Figure 1: MOXcatter tag prototype.

Much of the energy in WiFi and most other existing wireless technologies are incurred during transmitting signals. Recently, backscatter communication has changed the landscape [2]. A reader in a backscatter communication system provides excitation signals, and a tag uses on-board circuits or programmable logic to modulate and reflect an excitation signal by controlling the matching impedance of its antennas. The reader, after receiving the reflected signal, can then decode the tag data accordingly. The implementation can be highly energy efficient given that the data from the tag is carried by reflected, rather than proactively generated, signals; the tag can also harvest energy from the excitation signals, which is sufficient for powering the computation and transmission units on-board. As such, a tag can simply be battery-free. There have been significant studies on backscatter communications, in particular, Radio Frequency IDentification (RFID), which is now available in the mass market with extremely low cost. Despite reflecting dedicated RFID signals, recent studies suggest that a tag can indeed reflect a broad spectrum of signals, e.g., from cellular, wireless TV, WiFi, etc., offering a novel low-cost and low-energy solution to communicate with these wireless interfaces [3, 6, 7, 9, 14–16, 28, 29].

Given that many of the IoT devices are expected to be WiFi-compatible but do not necessarily need the full set of WiFi functionalities, WiFi backscatter has received great interest. Passive Wi-Fi [5] has enhanced the functionality of backscatter systems to work with off-the-shelf WiFi devices. Yet it requires an additional plugged-in device to generate the excitation signal. Since 2009, MIMO has been incorporated in all the mainstream WiFi standards, i.e., 802.11n/ac/ad. To use the ambient WiFi signals for excitation, it is necessary to consider the MIMO-based signals, which are ubiquitous today. For MIMO with single-stream and multi-stream modes, an AP can dynamically select between the two modes based on channel quality [42, 43]. Such advanced features have yet to be explored, in particular, *spatial multiplexing*. As a matter of fact, Inter-Technology Backscatter [6], HitchHike [8], and FreeRider

MobiSys'18, June 10–15, 2018, Munich, Germany

ACM ISBN 978-1-4503-5720-3/18/06…$15.00
https://doi.org/10.1145/3210240.3210329

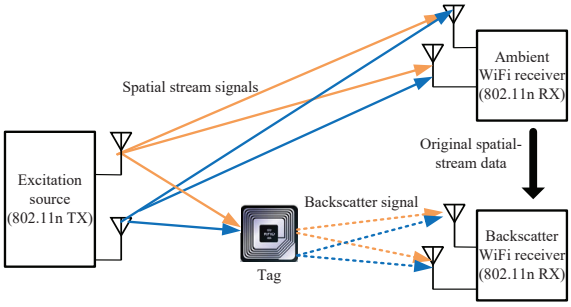


Figure 2: MOXcatter communication using commodity MIMO WiFi.

[30], though being compatible with commercial WiFi devices, all deal with single-antenna signals; Wi-Fi Backscatter [3] and FS-Backscatter [7] use the CSI/RSSI information rather than spatial streams to convey the tag information.

To explore the full potential of the latest WiFi standards, we introduce MOXcatter, a spatial-stream backscatter system using commodity multi-antenna WiFi. MOXcatter can embed tag data on ambient spatial streams, which can be decoded using commercial off-the-shelf MIMO WiFi devices. The basic process is illustrated in Figure 2, in which spatial multiplexing WiFi is used as the RF excitation source. The backscatter tag uses the spatial stream data to carry the tag information; a backscatter WiFi receiver (a commodity WiFi device) then decodes the tag information by comparing the backscattered data and the original data from the ambient WiFi receiver (another commodity WiFi device).

The key contribution of this paper is the design, implementation and evaluation of the backscatter system working with MIMO WiFi, which entails the following challenges:

- For a single-stream signal, an OFDM symbol contains consecutive bits in the original data bit stream (e.g., 26 consecutive bits in an OFDM symbol of 802.11n single-stream 6.5Mbps). For a multi-stream signal, however, the bits contained in an OFDM symbol are not consecutive due to stream parsing, and hence tag modulation on individual OFDM symbols would not be demodulated.
- There is a need for synchronization between the tag information and the spatial stream data. Tag modulation must operate on specific data fields in a packet, otherwise the tag data's bit error rate would significantly increase.
- A tag needs to seamlessly work with different excitation signals. Although it is possible to include multiple modulation modules in a single tag for different excitation signals (e.g., single-stream signals and multi-stream signals), the selection between them needs to be automated.

To address the above issues, MOXcatter incorporates the following novel designs:

- We design a modulation scheme that allows a MOXcatter tag to convey information by changing the phase of OFDM symbols in spatial streams. It conveys a '0' or '1' bit using either a 0° or a 180° phase change. In particular, such a phase change operates on individual OFDM symbols to enable a single-stream packet with multiple bits of tag data, while it operates on the whole data fields to enable a multi-stream packet with only one bit of tag data.
- We design a customized decoding process that extracts the phase change information by comparing the backscattered data with the original data. It searches each all-zero (or all-one) sequence from a bit stream that is obtained by an XOR of the backscattered data and the original data. The sequence length is equal to the number of bits contained in an OFDM symbol. The all-zero (or all-one) sequence represents the 0° (or 180°) phase change by the tag.
- We use a WiFi signal detector and synchronous digital logic circuits in the tag to receive the control signals and automatically select the corresponding backscattering circuits for different excitation signals.

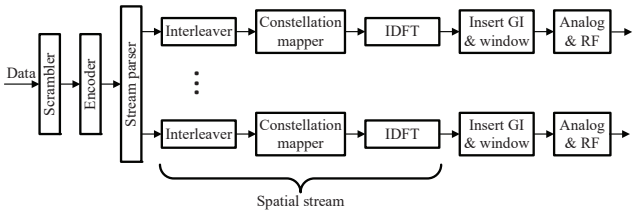


Figure 3: MIMO WiFi transmitter.

We have built a MOXcatter tag prototype, which integrates an FPGA and analog RF front-end circuits, as shown in Figure 1. Our experiments show that MOXcatter achieves up to 50 Kbps throughput for single-stream signals and up to 1 Kbps for double-stream signals with a communication range (tag-to-RX) up to 14 meters. We discuss the tradeoff between the throughput and the implementation complexity, as well as possible enhancements toward high throughput, particularly for the multi-stream case. We also showcase the applicability of our design through a sensor system using only MOXcatter tags and commodity WiFi devices.

2 PHY OF ADVANCED WIFI NETWORKS

Spatial multiplexing radio based on MIMO-OFDM has been the dominant PHY air interface in such advanced WiFi standards as 802.11n/ac/ad/ax, and is also the cornerstone of our work. A typical spatial stream transmitting process is shown in Figure 3. It includes a series of transform operations [31–34], which will be explored in our MOXcatter design. We start from a brief introduction on them here and, in the next section, we will illustrate how to modulate the tag data with phase change to make a backscattered packet still a legit WiFi packet.

Scrambling: In digital communications, if long sequences of all 0 (or 1) frequently occur, they will have a negative impact on the symbol synchronization at the receiver. Therefore, scrambling is introduced to break the long runs of the repeated data, and to change the statistics of the transmitted signal into an approximation of a white noise. To do so, a WiFi transmitter uses a frame-synchronous scrambler to generate a fixed length bit sequence to scramble the data field. The original data bits are placed in a bitstream and put into an XOR gate one by one, which can be denoted as follows:

$$ScrambledData = OriginalData \oplus ScramblerOut \quad (1)$$

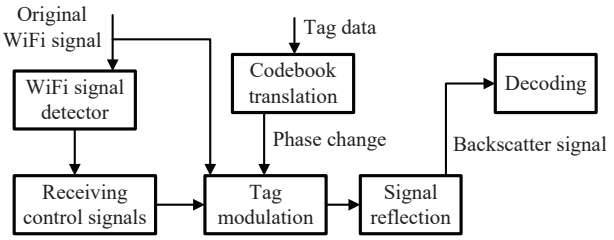


Figure 4: Overview of system design.

Convolutional Encoding: A convolutional code is not only correlated with the current input bit but also with the previous input bits. Convolutional encoding has low delay and fits serial data transmission well. When k bits information is encoded to n bits, the coding rate R is k/n. Different speeds of 802.11n MIMO-OFDM correspond to different subcarrier modulations and coding rates, e.g., 6.5 Mbps BPSK with $R = 1/2$, 19.5 Mbps QPSK with $R = 3/4$, and 65Mbps 64-QAM with $R = 5/6$.

Stream Parsing: Spatial multiplexing WiFi introduces a stream parser that divides the outputs of the encoder into blocks of s bits and sends the blocks to different interleavers. For example, the block size is $s = 1$ for both 802.11n single stream BPSK-subcarrier modulation and double streams BPSK-subcarrier modulation. The encoded data sequence sent to an interleaver is called a *spatial stream*. The stream parser assigns the consecutive blocks of bits to different spatial streams in a round robin manner. The multiple spatial streams are then transmitted independently by different antennas on the same channel simultaneously. In this paper, we utilize the spatial stream data to convey information, and in Section 3 we will illustrate how the tag-modulated spatial stream data is decoded with existing commodity WiFi radios.

Interleaving: Bit errors usually happen in some serial bits. In order to overcome the serial bit errors or burst errors, interleaving is used to rearrange data in a noncontiguous manner and change a long string of error bits into some short strings of error bits that can be corrected by forward error correction.

Constellation Mapping: It refers to the OFDM subcarrier modulation methods, including BPSK, QPSK, 16-QAM, 64-QAM, and 256-QAM. For each subcarrier, one or more coded data bits are mapped into a point (complex) on the constellation according to the corresponding modulation methods. The output of the constellation mapper is used to generate the input of an Inverse Discrete Fourier Transform (IDFT) of each spatial stream.

Inverse Discrete Fourier Transform: The 802.11g OFDM modulation uses 52 subcarriers, of which 48 are for data and 4 are for pilot signals. The 802.11n 20 MHz HT format uses 56 subcarriers for data and involves an IDFT process that generates the OFDM symbols. After IDFT, the OFDM symbols are ready to multiplex the carrier wave for transmission.

3 MOXCATTER DESIGN

An architectural view of our MOXcatter is shown in Figure 4. In this section, we will introduce the design of its key modules, including signal detecting, tag modulating, and information decoding, as well as control signaling.

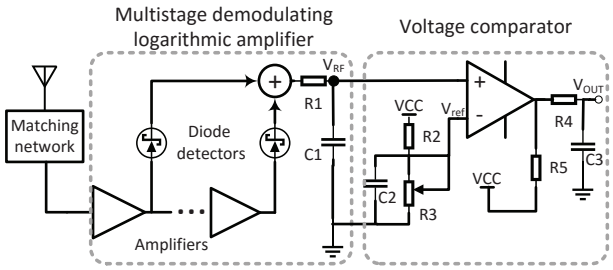


Figure 5: Design of WiFi signal detector circuit.

3.1 WiFi Signal Detector

The signal detector is a front-end analog signal circuitry between the antenna and the FPGA module on a MOXcatter tag's circuit board. It converts the original incoming WiFi power signal into a usable binary high/low voltage signal for the FPGA's digital circuitry, where a high voltage outputs if a WiFi signal is detected, and a low voltage otherwise. As shown in Figure 5, its design includes two parts: a multistage demodulating logarithmic amplifier and a voltage comparator. The multistage amplifier circuit works as the WiFi power detector, whereas each amplifier stage is associated with a diode detector. The multistage outputs are summed up and passed to a low pass filter (consisting of $R1$ and $C1$). The output V_{RF} is proportional to the input signal's amplitude. Let dBm_{RF} be the strength of the WiFi signal received by the tag antenna. V_{RF} can be formulated as follows:

$$V_{RF} = K \cdot dBm_{RF} \tag{2}$$

where K is a constant scaling factor in the operating range between the intercept and saturation points.

The voltage comparator circuit is designed to filter the noise. We have observed in our experiments that the WiFi signal level is usually much higher than the noise level, e.g., 30 dB. Accordingly, we can use a threshold voltage to obtain the detected WiFi signal. As shown in Figure 5, a sliding rheostat $R3$ is used to set the threshold V_{ref}. If the comparator's input $V_{RF} > V_{ref}$, V_{OUT} will be '1', and '0' otherwise.

Figure 6 demonstrates two traces of voltage V_{RF}, which were captured during two WiFi transmission experiments with commodity WiFi adapters. In each experiment, 2000 packets were sent, each of 556 Bytes, and two different modulation methods have been used: 802.11b DQPSK and 802.11n MIMO-OFDM (2×2, BPSK subcarrier), respectively. The results clearly show that the detector identifies useful patterns, including the beginning, the ending and the duration of the WiFi signals. The positive edge of the beginning and the negative edge of the ending can be used as trigger signals for the following FPGA circuits.

3.2 Tag Codebook for Spatial Stream signal

After the spatial stream signal has been detected, its OFDM symbols can convey the tag information using phase modulation. Let $Y_n = \{y_1^{(n)}, y_2^{(n)}, ..., y_m^{(n)}\}$ be the sequence of the complex coded data contained in the nth OFDM symbol, where m is the number of the coded data in an OFDM symbol, and $y_r^{(n)}$ represents the I/Q

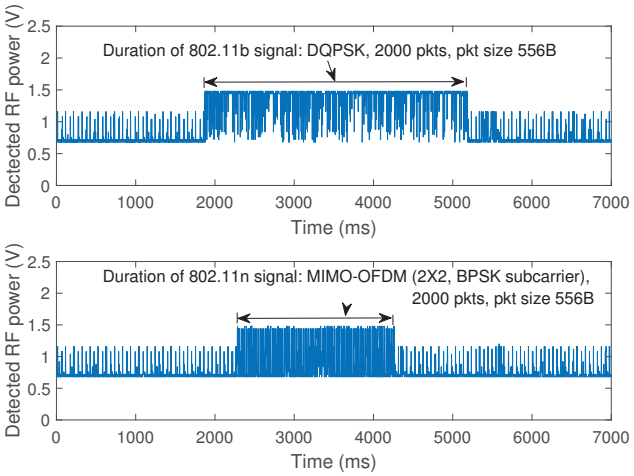


Figure 6: Comparison of signal durations using different modulation methods.

data on the subcarrier r of symbol n. $\mathbf{Y} = [Y_1, Y_2, ..., Y_l]'$ is hence the vector of the transmitted OFDM symbols.

MOXcatter's tag modulation follows a codebook to change the subcarrier phase by multiplying the OFDM symbol vector with a tag modulation vector $\Theta = [\theta_1, \theta_2, ..., \theta_l]$, where $\theta_i \in \{-1, 1\}$, $i \in \{1, ..., l\}$ for subcarrier modulation using BPSK, and $\theta_i \in \{e^{-j\frac{\pi}{2}}, e^{j\frac{\pi}{2}}, -1, 1\}$, $i \in \{1, ..., l\}$ for subcarrier modulation using QPSK, 16-QAM and 64-QAM. Therefore, the tag-modulated OFDM symbol matrix can be expressed as:

$$\mathbf{Y}^{tag} = \Theta \cdot \mathbf{Y}. \tag{3}$$

We can see that the tag modulation is a linear transform with respect to the I/Q data.

We next illustrate why a backscattered packet is still a legit 802.11n WiFi packet and why the operations (scrambling, convolutional coding, interleaving and IDFT) do not affect our tag modulation and demodulation.

Suppose we have a single stream at 6.5 Mbps with 802.11n. Each OFDM symbol is generated from the complex numbers of 52 data subcarriers and the values of 4 pilot subcarriers. For each OFDM symbol, the pilot values are inserted into the complex number sequence to compose a new sequence. The new sequence X_n is used as the input of IDFT, and we have $X_n = Inst(Y_n) = \{y_1^{(n)}, ..., y_7^{(n)}, p_{-21}, y_8^{(n)}, ..., y_{20}^{(n)}, p_{-7}, y_{21}^{(n)}, ..., y_{32}^{(n)}, p_7, y_{33}^{(n)}, ..., y_{45}^{(n)}, p_{21}, y_{46}^{(n)}, ..., y_{52}^{(n)}\}$, where $p_{-21}, p_{-7}, p_7, p_{21}$ are the pilot values. The IDFT is formulated as follows:

$$x_n(t) = IDFT\Big[X_n(k)\Big] = \frac{1}{K}\sum_{k=0}^{K-1} X_n(k) e^{j\frac{2\pi}{K}kt} \tag{4}$$

where $t = 0, 1, ..., K-1$ and K is the number of the subcarriers. $x_n(t)$ is the discrete-time signal of symbol n and is ready to multiply the carrier for transmission. The pilot values are for synchronization only, which do not affect the demodulation using DFT at the receiver. Hence, the above transform is linear. Given that the phase change synchronizes with and operates on the nth OFDM symbol, the tag

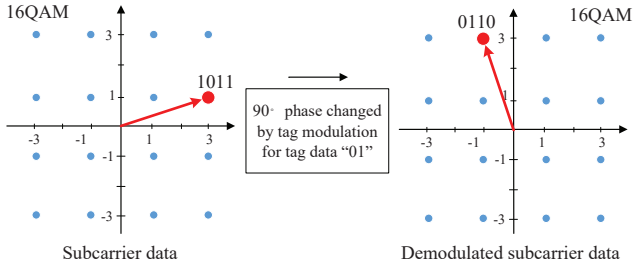


Figure 7: An illustration of why the tag-modulated subcarrier data still fall in the constellation of QAM and hence can be demodulated by a commodity WiFi device.

modulation satisfies

$$\theta_n \cdot IDFT\Big[Inst(Y_n)\Big] = IDFT\Big[Inst(\theta_n \cdot Y_n)\Big]. \tag{5}$$

Equation (5) is performed on the channel for receiving the backscattered signal (i.e., a fixed frequency shift from the excitation signal's channel); when tag modulation changes the phase of the OFDM symbol waveform, it actually changes the I/Q data. We should ensure that the changed data is still valid for subcarrier demodulation at the receiver.

Let $C_n = \{c_1^{(n)}, ..., c_q^{(n)}\}$ be the original data sequence contained in the nth OFDM symbol, where q is the number of data bits per symbol under a specific modulation . Let $D_n = \{d_1^{(n)}, ..., d_q^{(n)}\}$ be the scrambled data sequence. Let $S_n = \{s_1^{(n)}, ..., s_q^{(n)}\}$ be the sequence generated by the scrambler. According to Equation (1), D_n can be expressed as:

$$D_n = S_n \oplus C_n \tag{6}$$

Since the convolutional coding and interleaving are both linear, let $\theta_n \cdot x_n = V(S_n \oplus C'_n)$, where C'_n is the data sequence after the tag modulation and $V(\cdot)$ is the linear transform including convolutional coding, interleaving and IDFT.

At the receiver, the data is demodulated as follows:

$$V^{-1}\Big(\theta_n \cdot x_n\Big) \oplus S_n = S_n \oplus C'_n \oplus S_n = C'_n. \tag{7}$$

According to the constellation analysis in Figure 7 and the tag codebook in Table 1, C_n and C'_n have the same codebook. For QPSK, 16QAM and 64QAM in Table 1, only the four phases 0°, 90°, 180° and 270° ensure that the backscattered I/Q data still fall in the same constellation, thereby carrying only two bits of the tag information.

For 802.11n multiple streams, each antenna independently transmits a spatial stream at the same frequency channel. Suppose the sender and the receiver are two WiFi devices, each having N antennas. $[\mathbf{x}_1, ..., \mathbf{x}_N]$ represents the original spatial stream signals generated from antenna $1, 2, ...N$, respectively. $[\mathbf{x}_1^{tag}, ..., \mathbf{x}_N^{tag}]$ represents the backscattered signals. $[\mathbf{Z}_1, ..., \mathbf{Z}_N]$ represents the received backscattered signals, which can be formulated as

$$\begin{bmatrix} \mathbf{Z}_1 \\ \vdots \\ \mathbf{Z}_N \end{bmatrix} = \begin{bmatrix} h_{11} & \cdots & h_{1N} \\ \vdots & \ddots & \vdots \\ h_{N1} & \cdots & h_{NN} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1^{tag} \\ \vdots \\ \mathbf{x}_N^{tag} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_1 \\ \vdots \\ \mathbf{n}_N \end{bmatrix} \tag{8}$$

Table 1: Tag Codebook for spatial stream signal.

Subcarrier modulation	Subcarrier phase changed by tag	Tag data
BPSK	0°	0
	180°	1
QPSK, 16QAM, 64QAM	0°	00
	90°	01
	180°	10
	270°	11

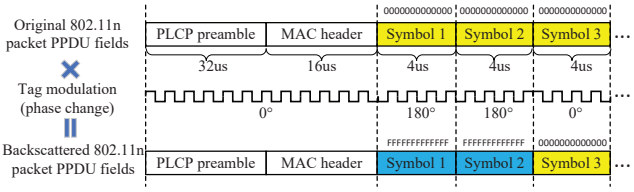


Figure 8: Tag modulation: multiplying the original 802.11n signal by a tag modulation signal (using phase change of a square wave to represent tag information).

where $H = [h_{ij}]^{N\times N}$ is the time-domain MIMO channel and $[\mathbf{n}_1,...,\mathbf{n}_N]'$ is a noise vector. The matrix H is related to channel properties (e.g., channel attenuation, carrier phase offset, and propagation delay). Since an 802.11n transmitter assigns the coded data to different spatial streams in a round robin manner, the tag modulation using phase change is no longer a linear transform with respect to the coded data. When using stream parsing in the 802.11n MIMO, any OFDM symbol in $\mathbf{x}_i^{tag}$ of Equation (8) would contain non-consecutive data bits compared with the original data bit stream. In our design, a phase change by the tag modulation cannot be decoded from non-consecutive bits. Hence, the phase change does not operate on individual OFDM symbols for the multi-stream signals. Yet we can still use phase change 0° and 180° to embed one bit tag data on each packet (i.e., the phase change operating on the whole PPDU data fields). The backscattered 802.11n multi-stream PPDU data fields do not change if the tag modulation uses a 0° phase change, unless a 180° phase change is used.

3.3 Tag Modulation

According to the analysis in Subsection 3.2, if an OFDM symbol's phase is changed by a MOXcatter tag following the codebook in Table 1, the backscattered signal can be decoded by commodity WiFi devices. MOXcatter tag uses the phase modulation that multiplies the original 802.11n signal by a phase-varying square wave signal. The phase of the square wave changes when the tag data changes. As illustrated in Figure 8, the tag does not change the phase of the preamble in the PHY Layer Convergence Procedure (PLCP) and the MAC header fields in a PLCP protocol data unit (PPDU), so as to ensure that each backscattered packet is decodable. The symbol phase is then changed according to the tag data and the codebook. The signal multiplication is done by using an RF switch, whose on-off control signal is a phase-varying square wave.

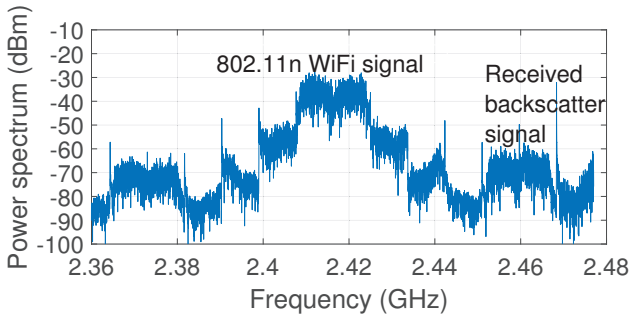


Figure 9: Spectrum of the original 802.11n WiFi signal and the backscattered signal using 50MHz frequency shift from 2.417GHz (Channel 2) to 2.467GHz (Channel 12).

3.4 Modulated Signal Reflection

MOXcatter tag uses a frequency f_t square wave signal to control the on-off frequency of the RF switch. f_c is the carrier center frequency of the 802.11n signal. Let $\omega_t = 2\pi f_t$, $\omega_c = 2\pi f_c$, and $\alpha_{base}(t)$ be the baseband waveform of the 802.11n signal. The square wave can be formulated as $M_{tag}(t) = \frac{4}{\pi}\sum_{n=1}^{\infty}\frac{sin((2n-1)\omega_t t)}{2n-1}$. Hence, the backscattered signal, $\beta(t)$, can be formulated as

$$\beta(t) = \alpha_{base}(t)e^{jw_c t}M_{tag}(t). \tag{9}$$

Let $F_{base}(\omega)$ be the Fourier transform of $\alpha_{base}(t)$, and $F(\omega)$ be the Fourier transform of $\beta(t)$. We have

$$F(\omega) = \sum_{n=1}^{\infty}\frac{2j}{\pi(2n-1)}\Big(F_{base}(\omega-\omega_c+(2n-1)\omega_t) - F_{base}(\omega-\omega_c-(2n-1)\omega_t)\Big). \tag{10}$$

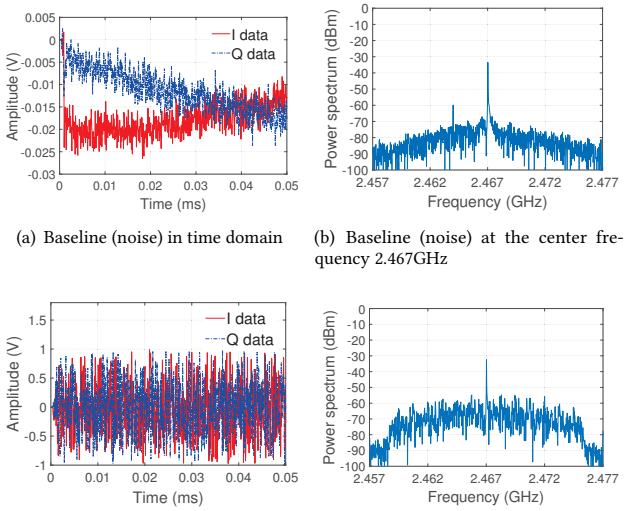
The backscattered signal is received at the frequency spectrum $\frac{2}{\pi}|F_{base}(\omega-\omega_c-\omega_t)|$. Figure 9 shows an example of the spectrum, and Figure 10 further shows the captured time domain and frequency domain backscattered signals, as compared with the noise signal baseline.

3.5 Tag Information Decoding

The backscattered signal is received and decoded by a commodity WiFi device with multiple antennas, e.g., with an 802.11n NIC. Since a MOXcatter tag uses frequency shift to forward the backscattered 802.11n packets, the original 802.11n packets and the backscattered 802.11n packets will be received at two different frequencies with a fixed difference. They will be used together to decode the tag information, following two steps.

In the first step, we use the decoder proposed by [8]. The original 802.11n packets are captured and decoded at the center frequency 2.417GHz (Channel 2), and the backscattered 802.11n packets are captured and decoded at the center frequency 2.467GHz (Channel 12). Let $\mathbf{Q} = \{Q_1,...,Q_n\}$ be the original 802.11n data, where Q_i is the data bits of the ith OFDM symbol. Let $\mathbf{B} = \{B_1,...,B_n\}$ be the backscattered 802.11n data, where B_i is the data bits of the ith OFDM symbol. The processed raw data $\Gamma = \{\Gamma_1,...,\Gamma_n\}$ is obtained by performing the following XOR operation:

$$\Gamma_i = Q_i \oplus B_i \tag{11}$$



(a) Baseline (noise) in time domain

(b) Baseline (noise) at the center frequency 2.467GHz

(c) Backscattered I/Q signal in time domain

(d) Backscattered signal at the center frequency 2.467GHz

Figure 10: Backscattered WiFi signal captured at the center frequency 2.467GHz. The original WiFi signal is sent from the center frequency 2.417GHz and reflected with 50MHz frequency shift.

where $\Gamma_i, i \in 1,...,n$ is the processed raw data from the ith OFDM symbol. The length of Γ_i depends on the modulation methods, e.g., $|\Gamma_i| = 24$ for 802.11g OFDM BPSK subcarrier coding rate $\frac{1}{2}$, $|\Gamma_i| = 26$ for 802.11n MIMO-OFDM single stream BPSK subcarrier, and $|\Gamma_i| = 52$ for 802.11n MIMO-OFDM double streams BPSK subcarrier. The codeword translation method in [8] decodes the tag information in the bit level and works for 802.11b signals only. Hence, another step is required to decode the tag information in the OFDM-symbol level in our system.

The second step is to search and decode the specific sequence of a fixed-length $|\Gamma_i|$. Take BPSK subcarrier modulation as an example, in which the sequence is all-zero (or all-one). In the tag modulation, we can choose the number of the OFDM symbols that are used to carry one bit tag data. If we use two OFDM symbols to carry tag data 1, the corresponding time-domain waveform of the two OFDM symbols is changed by a 180 degree phase shift. When decoding the tag data, we use a decoding window whose length equals to the number of OFDM symbols for a tag data bit. An observation in our measurement is that bursty bit errors often occur near the change from a long sequence of all-zero (all-one) bits to a long sequence of all-one (all-zero) bits, i.e. phase change from 0 (180) degree to 180 (0) degree. This may lead to decoding failure when using one OFDM symbol for a tag data bit. For a single spatial stream using the BPSK-subcarrier modulation, if we use two OFDM symbols for a tag data bit, there is always an all-zero (or all-one) sequence of length $|\Gamma_i|$ contained in a decoding window. We can use such a sequence as the valid data to decode a tag data bit. Let Λ_j be an all-one sequence of length $|\Gamma_i|$ for the jth tag data bit, and $\Phi_j = [\Gamma_i, \Gamma_{i+1}]$, where $i = 2j - 1$ and j is a positive integer. The

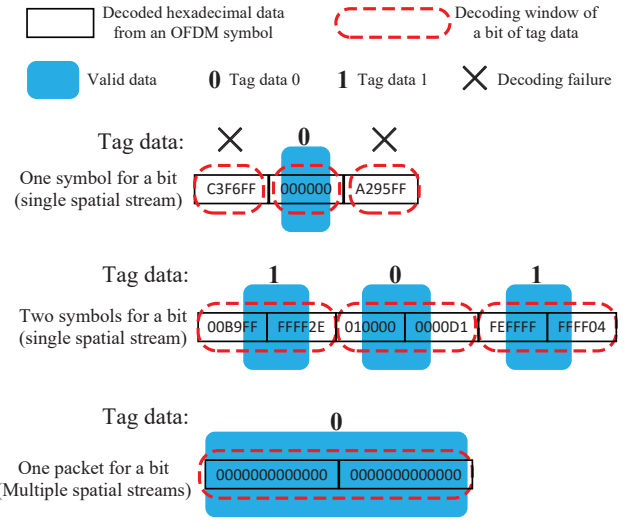


Figure 11: Tag data decoding using a decoding window and a search for fixed-length all-zero (all-one) sequences.

searching algorithm for an all-one sequence is as follows:

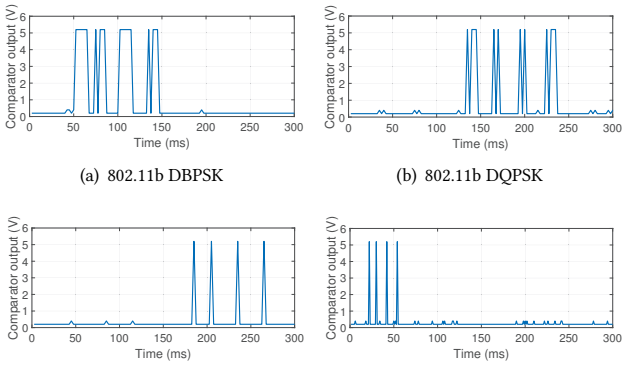
$$\eta* = \underset{\eta \in \{1,\ldots,|\Gamma_i|\}}{\operatorname{argmax}} \left| \sum_{k=0}^{|\Gamma_i|} \Lambda_j[k]\Phi_j[\eta + k] \right| \tag{12}$$

where $\eta*$ is the starting position of the valid data sequence. For searching an all-zero sequence, we only need to change argmax to argmin in Equation (12).

Figure 11 is an illustrative example, where a tag transmits a data sequence "101" using a single spatial stream signal. For the BPSK-subcarrier modulation used for 802.11n, the tag data decoding may fail when we use one symbol for one bit of the tag data; using a decoding window of length $2|\Gamma_i|$, the tag data can be successfully decoded. Only the first all-zero (or all-one) sequence of length $|\Gamma_i|$ in a window is used as the valid data for decoding. For the 802.11n MIMO-OFDM multi-stream BPSK-subcarrier modulation, our analysis in Subsection 3.2 indicates that we can only embed one bit tag data in the data fields of each packet. For the double-stream 802.11n data shown in Figure 11, if an all-zero sequence can be decoded from a backscattered packet's data fields, then it represents tag data 0, and 1 otherwise.

3.6 Control Signals

The output of the WiFi signal detector can measure the durations of a WiFi packet, which can be used to form different control signal patterns. This enables MOXcatter to receive the control signals from the excitation source and work in ambient WiFi environments with different excitation signals. Figure 12 shows the trace of four packets (each of 2024 Bytes) that were captured from the output of the voltage comparator. We use four different types of the 802.11 PHY to generate the control signals. When detecting the control signals, the voltage comparator outputs different patterns, corresponding to different excitation signals. The duration of a packet can be accurately measured by the FPGA's digital circuit. For example, if 2024 Bytes data are transmitted by one packet, the 802.11b



(a) 802.11b DBPSK (b) 802.11b DQPSK

(c) 802.11g OFDM 6Mbps BPSK subcarrier (d) 802.11n 2 × 2 MIMO-OFDM 13Mbps BPSK subcarrier

Figure 12: Signal detector's response to four consecutive packets (2024 Bytes packet size): the voltage comparator (in Figure 5) uses RF power detector output as its input and shows different output patterns, corresponding to different 802.11 PHY types.

1Mbps takes 16540μs, the 802.11g 6Mbps takes 2720μs, and the 802.11n double streams 13Mbps takes 671μs. The packet duration is calculated using a counter and the positive edges of a clock. Such information can then be used by the tag. In particular, we use the average duration of four consecutive packets as the condition to trigger the corresponding backscatter circuits. Increasing the number of recognizable types of the excitation signals is possible, though will be under the constraints of the processing speed and resources offered by the FPGA.

4 MOXCATTER IMPLEMENTATION

4.1 Detector Circuits

We have implemented the WiFi signal detector by connecting essential periphery circuits to the main integrated circuits. Following the design in Figure 5, the multistage demodulating logarithmic amplifier uses an AD8313 connected with a matching network and a low-pass filter. The voltage comparator uses a TLV3501, a threshold voltage tuning circuit and a low-pass filter. The threshold is set to 1.25V by tuning the sliding rheostat.

4.2 FPGA Digital Circuits

The tag modulation circuits and the control signal receiving circuits are implemented in a XILINX Spartan XC3S500E-4PQ208 FPGA. Figure 13 shows the main components of the tag modulation circuits. When WiFi signals are detected, the detector's output has a positive edge, which can trigger the tag modulation and also the counter. If the signal duration is longer than the threshold of a packet duration, the LESSCONSTANT:1 block would change the state of the FDPE flip-flop. Packet synchronization uses the first detected positive edge signal after four consecutive packets for control signals. The counter then continues to count the durations of each field in a PPDU frame. For 802.11g signals, the durations of the PLCP preamble, the MAC header, and each OFDM symbol

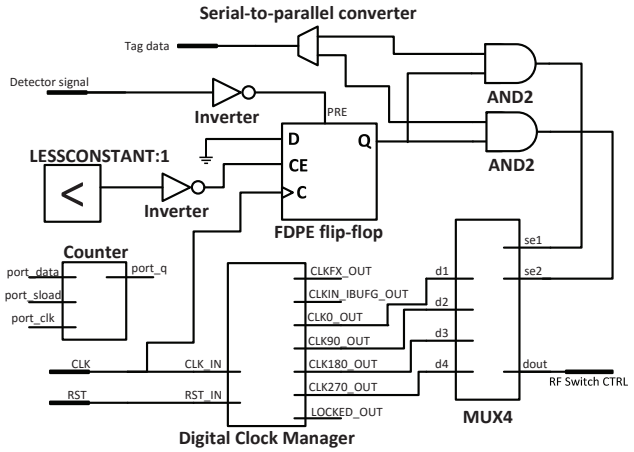


Figure 13: FPGA's digital circuit for tag modulation.

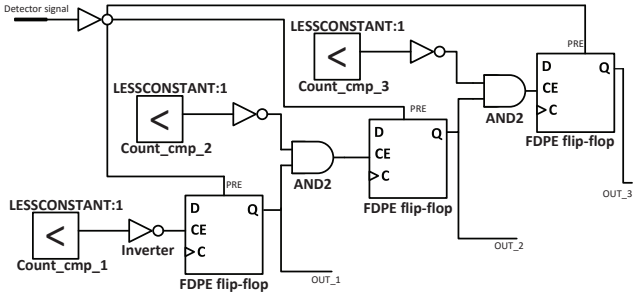
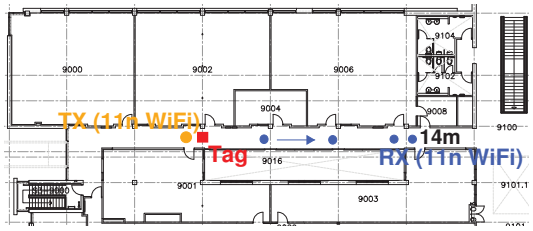


Figure 14: FPGA's digital circuit for receiving control signals.

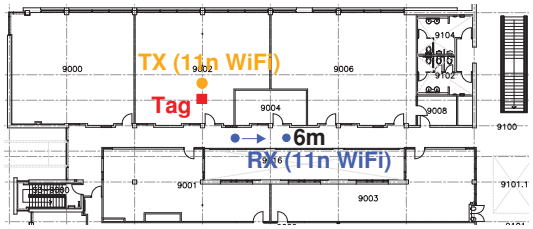
are 20μs, 16μs, and 4μs, respectively; For 802.11n signals, the durations are 32μs, 16μs, and 4μs, respectively. The square wave for the tag modulation is generated by the Digital Clock Manager (DCM) module of XILINX FPGA. The four outputs `CLK0_OUT`, `CLK90_OUT`, `CLK180_OUT` and `CLK270_OUT` have phase changes 0°, 90°, 180° and 270°, respectively. The tag data outputs connect to the selection pins `se1` and `se2` of the multiplexer. The multiplexer output is selected among the four outputs of the DCM, following the codebook in Table 1, and is used to control the ADG902 RF switch.

Figure 14 shows the FPGA's digital circuit for receiving control signals, which consists of LESSCONSTANT:1 blocks, FDPE flip-flops, inverters, and AND2 gates. It works in three stages and can differentiate three different excitation signals (802.11b 2Mbps, 802.11g 6Mbps and 802.11n DS 13Mbps in our implementation). The three LESSCONSTANT:1 blocks count if the packet duration falls in the range of the three stages (Count_cmp_3 > Count_cmp_2 > Count_cmp_1). If the packet duration falls in Stage one, OUT_1 becomes to 0, and both OUT_2 and OUT_3 are 1. This is because the two AND2 outputs are zeros, which do not change the state of the two FDPE flip-flops.

We can also reserve FPGA's resources to extend the MOXcatter's capability. For instance, free FPGA pins can be connected to such periphery circuits or components as accelerometers, humidity and

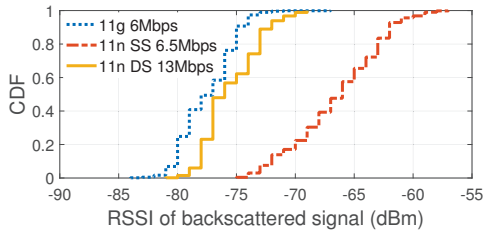


(a) Line-of-sight

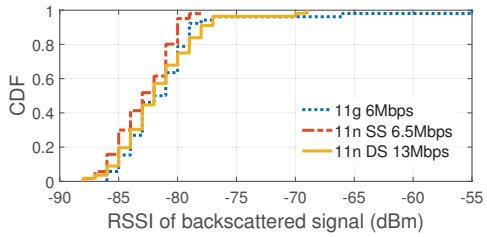


(b) Non-line-of-sight

Figure 15: Experiment scenarios for evaluating MOXcatter performance changed with tag-to-RX distance: floor plan of line-of-sight and non-line-of-sight scenarios.



(a) Distance = 2m



(b) Distance = 6m

Figure 17: Backscattered signal strength in non-line-of-sight experiments (TX-to-tag distance is 0.3m).

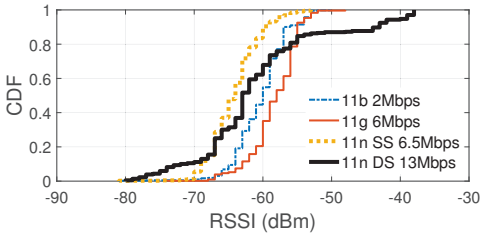


Figure 18: Backscattered signal strength.

temperature sensors. The drive circuits for the external components can be implemented in the FPGA chipset as well.

4.3 Packet Transmission and Reception

In our implementation, both the excitation signal transmitter and the backscattered signal receiver are commercial PCs each equipped with a Qualcomm Atheros AR938x wireless network adapter. We use the 802.11 a/b/g/n/ac wireless network analyzing tool CommView for WiFi [35]. This software tool can build a WiFi packet of any 802.11 frame types, self-defined MAC header and data fields, and it can control a WiFi adapter to send packets with specific modulation methods. It can also work as a traffic parser that captures packets, decodes packet data, and records important information such as signal strength. The transmitter uses the tool to generate the excitation signals, and the receiver uses it to obtain the backscattered data. We then implement the searching algorithm introduced in Subsection 3.5 to decode the tag information from the backscattered data.

5 EVALUATION

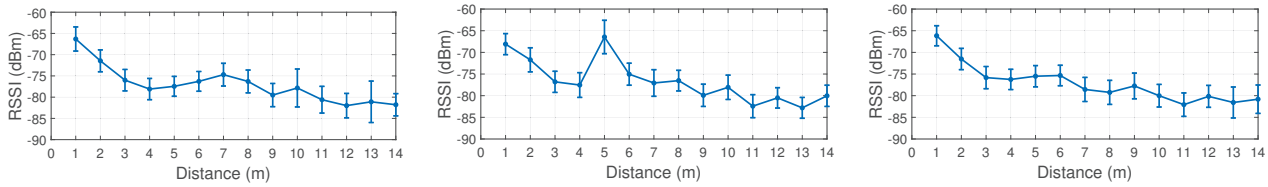
In this section, we evaluate the MOXcatter performance with both line-of-sight and non-line-of-sight experiments. We start from measuring the backscattered signal strength with different communication distances and physical layer specifications. We then examine the bit error and throughput of the decoded tag data.

5.1 Experiment Setup

We conduct our experiments in a 33m×15m indoor area. Figure 15 shows the floor plan and the configurations for both line-of-sight and non-line-of-sight scenarios. In the line-of-sight scenario, we place the 802.11n WiFi transmitter/receiver and the MOXcatter tag in a corridor. At the beginning, all of them are located in the same position, which is outside of the office door, and we then move the receiver along the corridor to increase its communication distance to the MOXcatter tag. In the non-line-of-sight scenario, we place the WiFi transmitter and the MOXcatter tag inside an office, and the receiver outside of the office door. We then move the receiver along the corridor starting from a 2m distance to the tag. For each experiment configuration, we measure the Received Signal Strength Indicator (RSSI), the Bit Error Rate (BER) and the throughput. In the measurements, the MOXcatter tag uses a 5V external power supply, and the TX-to-tag distance is 0.3 m.

5.2 Backscattered Signal Strength

We first measure the backscattered signal's RSSI in the line-of-sight scenario. We use four different physical layer specifications for the WiFi excitation signals, including 802.11b QPSK 2Mbps, 802.11g OFDM BPSK-subcarrier 6Mbps, 802.11n MIMO-OFDM Single-Stream (SS) BPSK-subcarrier 6.5Mbps, and 802.11n MIMO-OFDM Double-Stream (DS) BPSK-subcarrier 13Mbps. The excitation signal is generated at Channel 2 (2.417GHz), and the backscattered signal is



(a) 802.11g 6Mbps (b) 802.11n SS 6.5Mbps (c) 802.11n DS 13Mbps

Figure 16: Backscatter RSSI changed with tag-to-RX distance in line-of-sight experiments (TX-to-tag distance is 0.3m).

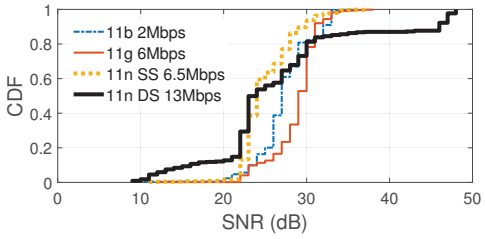


Figure 19: Ratio of backscattered signal level to noise level.

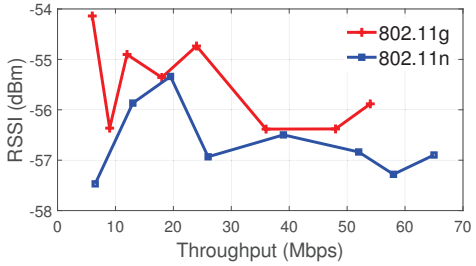


Figure 20: Backscattered signal strength vs. throughput.

received at Channel 12 (2.467GHz). We use the software tool CommView for WiFi to analyze the backscattered packets, which are captured by a Qualcomm Atheros AR938x wireless network adapter. We record the per-packet signal level and noise level.

Figure 16 shows the backscattered signal strength with different distances between the MOXcatter tag and the 802.11n WiFi receiver in the line-of-sight scenario. For all the three cases (802.11g 6Mbps, 802.11n SS 6.5Mbps and 802.11n DS 13Mbps), the general trend is that the RSSI value decreases with an increase of the backscatter communication distance, though there are certain fluctuations. When the distance increases to 14m, the RSSI values become lower than -80dBm for all the three cases.

In the non-line-of-sight experiments, Figure 17 plots the cumulative distribution function (CDF) of the RSSI, and it shows that the backscattered 802.11n signals have a higher signal strength than the backscattered 802.11g signals at a 2m distance, but the signal strength sharply decreases to a very low level (less than -80dBm) when the distance increases to 6m.

Figure 18 plots the CDF of the RSSI for the four types of WiFi signals at a 1m distance (tag-to-RX). First, a significant portion of the backscattered packets have a signal strength from -70dBm to -52dBm. More than 70% of the backscattered 802.11n DS packets and nearly all the backscattered 802.11b, 802.11g and 802.11n SS packets fall within -70~ -52dBm. Second, in the range from -70dBm to -58dBm, the 802.11g 6Mbps signal has the highest signal strength among the four types of WiFi signals. The 802.11b and 802.11g signals have higher strength than the 802.11n signals in this range.

Figure 19 shows the signal-to-noise ratio (SNR) of the backscattered signal. We can see that more than 50% of the backscattered 802.11n DS packets and nearly all the backscattered 802.11b, 802.11g and 802.11n SS packets have SNR values from 22dB to 33dB. 802.11b/g backscatter signals have a better SNR performance than 802.11n in the range of 22~28dB.

We also measure the backscattered signal strength with respect to the throughput of the WiFi excitation signal. We select the modulations that can achieve a throughput higher than 6Mbps to generate the WiFi excitation signals (e.g., 802.11n MIMO-OFDM SS 64-QAM-subcarrier 65Mbps). Figure 20 shows the backscattered signal's average RSSI values with respect to the original WiFi signals' throughput. The backscattered 802.11n signal has the maximum RSSI value at 19.5Mbps. When the throughput changes from 6Mbps to 65Mbps, the backscattered 802.11g signal has higher RSSI values than the backscattered 802.11n signal.

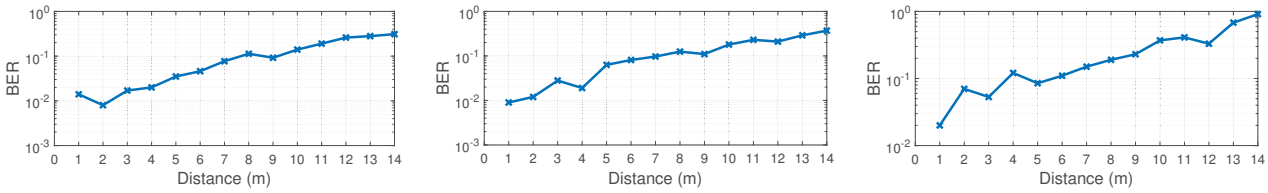
5.3 Bit Error Rate

We further measure the bit error rate (BER) in these backscatter experiments. Figure 21 shows the decoded tag data BER as a function of the distance between the MOXcatter tag and the 802.11n WiFi receiver in the line-of-sight experiments. Not surprisingly, the BER value generally increases with an increase of the distance. When the distance is less than 3m, it is below 0.06. It however increases sharply and becomes very high when the distance is over 14m, particularly for the backscattered 802.11n DS signals.

Figure 23 shows the decoded tag data BER in the non-line-of-sight experiments. The single-stream backscatter communication clearly has better performance than the double-stream case, though the working distances in both cases are shorter than the line-of-sight case.

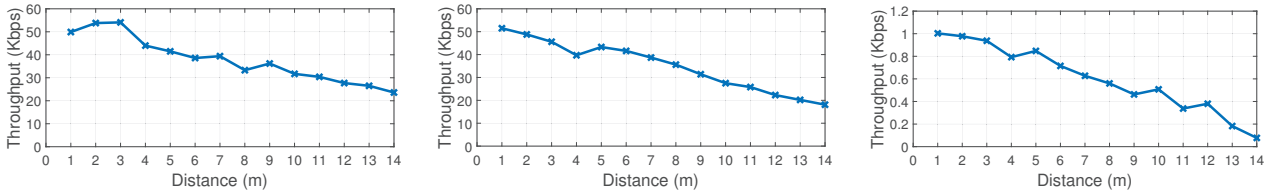
5.4 Throughput

We evaluate the backscatter communication throughput in both line-of-sight and non-line-of-sight experiments. We use the three different physical layer specifications, including 802.11g OFDM BPSK-subcarrier 6Mbps, 802.11n MIMO-OFDM SS BPSK-subcarrier 6.5Mbps, and 802.11n MIMO-OFDM DS BPSK-subcarrier 13Mbps, as excitation signals.



(a) 802.11g 6Mbps (b) 802.11n SS 6.5Mbps (c) 802.11n DS 13Mbps

Figure 21: Tag data bit error rate changed with tag-to-RX distance in line-of-sight experiments (TX-to-tag distance is 0.3m).



(a) 802.11g 6Mbps (b) 802.11n SS 6.5Mbps (c) 802.11n DS 13Mbps

Figure 22: Backscatter throughput changed with tag-to-RX distance in line-of-sight experiments (TX-to-tag distance is 0.3m).

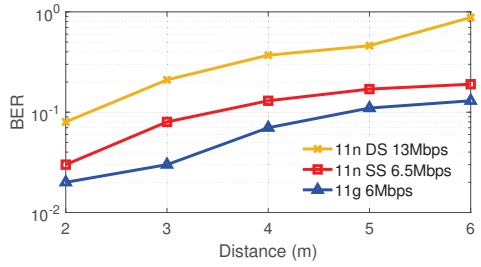


Figure 23: Tag data bit error rate in non-line-of-sight experiments (TX-to-tag distance is 0.3m).

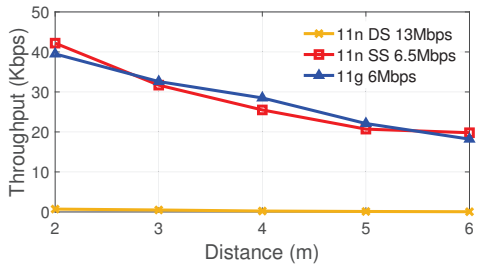


Figure 24: Backscatter throughput in non-line-of-sight experiments (TX-to-tag distance is 0.3m).

As shown in Figure 22, the throughput has a general downward trend with longer distance for all the three cases (with 802.11g OFDM BPSK-subcarrier 6Mbps, 802.11n MIMO-OFDM SS BPSK-subcarrier 6.5Mbps, and 802.11n MIMO-OFDM DS BPSK-subcarrier 13Mbps, respectively, as excitation signals). The backscattered 802.11n single-stream signals can achieve a higher than 40Kbps throughput when the distance is less than 6m. We can see in Figure 22(c) that the throughput is limited to less than 1Kbps for the case of double spatial streams. This is because we can only use the data fields of an 802.11n DS packet to carry one bit of tag data.

Figure 24 plots the throughput in the non-line-of-sight experiments. The single-stream backscatter communication has the throughput similar to the backscattered 802.11g signal, where the communication range is still limited to 6m.

6 APPLICATION CASE: A SENSING SYSTEM

Our MOXcatter can also be used in a wide range of IoT applications where 802.11n WiFi serves the hub for data exchange. The experimental results in Section 5 show that a MOXcatter tag can communicate with a commercial WiFi receiver from a distance up to 14 m for multi-stream signals, and from a distance over 14 m for single-stream signals. Within the communication range from 0 m to 3 m, the backscatter throughput can be up to 50Kbps. There is a great potential to build indoor environment monitoring systems for smart home design, and to support other sensing-based IoT applications such as health-care systems, indoor positioning systems and smart control systems.

To demonstrate the applicability, we have implemented a low power data-collecting sensor communication system using MOXCatter. As shown in Figure 25, the system consists of a DS18B20 digital thermometer, a MOXcatter tag prototype, and a sink node (an 802.11n WiFi adapter, which also serves as the excitation signal source). The communication is one-hop: the thermometer first collects temperature data and outputs it as the MOXcatter tag's data input; the data is then transmitted to the sink node via backscatter communication.

We implement the DS18B20's drive circuit module and the LED display's drive circuit module in the FPGA of the MOXcatter tag.

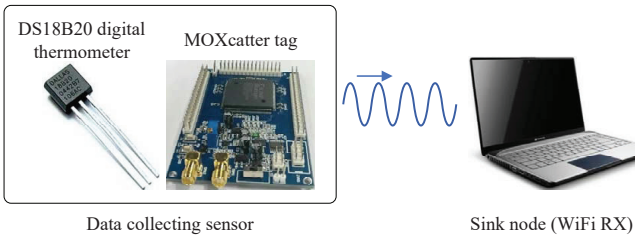


Figure 25: MOXcatter-based data collecting sensor communications.

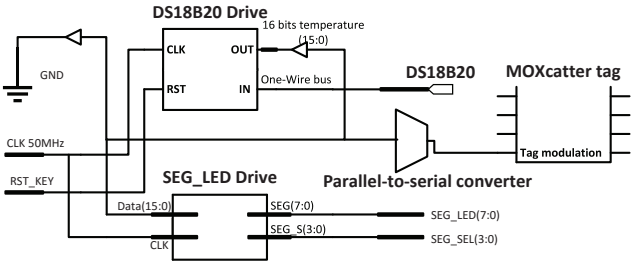


Figure 26: FPGA circuit design for MOXcatter-based sensor system.

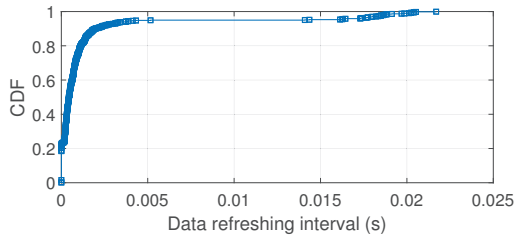


Figure 27: Distribution of temperature data refreshing intervals.

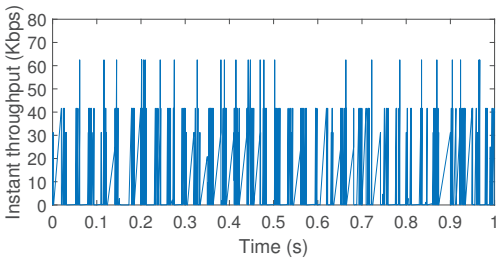


Figure 28: Instantaneous throughput of collecting temperature data.

As shown in Figure 26, the DS18B20's drive module outputs 16 bits parallel data, which are then converted to a serial data that generates a time-varying tag modulation signal. To connect the DS18B20 digital thermometer to the MOXcatter tag, the tag's circuit board employs a 1-Wire bus interface for sending/receiving the data from the thermometer. We use the 802.11n SS 6.5Mbps transmission as the excitation signal, and the data fields of each packet are set to all zero. During backscattering, the tag modulation embeds the sensed temperature data in a packet, which is then captured and decoded by the WiFi sink. Figure 27 shows the CDF of the data refreshing intervals at the sink node, which are calculated from the arriving time of each packet. We can see that over 90% of the interval values are less than 5ms. Figure 28 also shows the instantaneous communication throughput during one second.

7 RELATED WORK

Backscatter communication has received significant attention in recent years for its energy efficiency and low cost [3–29]. The excitation signal is usually generated by an external reader, which also decodes the backscattered signal and obtains the information from the tag. Besides using dedicated readers, recent studies have demonstrated that such ambient signals as FM, TV, cellular, and even WiFi, can be used for excitation. The latter is of particular interest given that WiFi is arguably the dominating indoor wireless communication technology.

Such pioneer studies as Wi-Fi Backscatter [3], BackFi [4] and Passive Wi-Fi [5] have advanced considerably in the design of WiFi-compatible backscatter systems. Wi-Fi Backscatter is the first communication system that connects backscatter tags to the Internet using WiFi devices. Its tag modulation works by reflecting or absorbing the original WiFi packets so that a backscattered packet's CSI/RSSI information is changed and the tag data can be decoded from the changed information. BackFi designs an IoT sensor for backscattering WiFi signals, together with customized radio circuits on a WiFi AP for decoding the backscattered signals. Passive Wi-Fi demonstrates that 10000x lower power is possible with WiFi backscatter than a standard WiFi communication. Recent studies further address many compatibility and deployment issues. Inter-Technology Backscatter [6] transforms wireless transmission from one technology to another; It can backscatter Bluetooth signals to generate decodable WiFi signals. FS-Backscatter [7] uses frequency shift to isolate the backscattered signal from the excitation signal and achieves robust decoding. It enables communication between commercial WiFi and Bluetooth radios. HitchHike [8] enables backscatter communication using only commodity WiFi devices. It embeds tag data on standard 802.11b packet data and decodes the tag data with any 802.11b receiver. FreeRider [30] uses the codeword translation [8] in a backscatter working with multiple commodity radios, including OFDM WiFi, ZigBee, and Bluetooth.

The WiFi technology evolves rapidly, and MIMO has been widely used in the new generation of WiFi devices based on 802.11n and beyond. It however has yet to be explored in the WiFi-based backscatter communication system. Initial attempts have mostly been confined to exploring the CSI/RSSI information rather than spatial streams [3, 7], supporting only single-antenna signals [6, 8, 30], or relying on specialized incompatible circuits [4]. Our MOXcatter for the first time demonstrates that spatial stream backscatter communication is possible with 802.11n and beyond. Specifically, it can embed tag information on the data fields of one or more 802.11n spatial streams and decode the tag data using a multi-antenna WiFi

device without any hardware modification. It is also backward-compatible, i.e., usable for backscattering 802.11b/g signals.

8 FURTHER DISCUSSION

8.1 Throughput: Tradeoff and Enhancement

It is well known that, with the same subcarrier modulation (e.g., BPSK for both 802.11n SS 6.5Mbps and 802.11n DS 13Mbps), the use of multiple streams increases the throughput for the communication between standard WiFi devices. This is naturally expected for WiFi-based backscatter communication, too. Our experimental evaluation however shows that our MOXcatter achieves up to 50 Kbps transmission for 802.11n single-stream 6.5Mbps signals and up to 1 Kbps for 802.11n double-stream 13Mbps signals. As discussed in Subsection 3.2, this is due to the stream parsing in the MIMO WiFi transmitter and the MOXcatter's tag modulation scheme. Our MOXcatter uses phase change on the WiFi packet data fields to modulate the tag information. The data fields consist of multiple OFDM symbols. When backscattering 802.11n single-stream signals, two consecutive OFDM symbols may experience different phase changes to carry different tag data. Intuitively, with such a symbol-level modulation, it is possible to increase the tag data rate by simply increasing the packet length. Unfortunately, it does not work for multi-stream signals. As introduced in Section 2, a MIMO WiFi transmitter uses a spatial stream parser, which assigns consecutive blocks of the encoded bits to different spatial streams in a round robin manner. An OFDM symbol of a multi-stream WiFi packet therefore does not correspond to consecutive bits of the original data. Since the MOXcatter's decoding scheme cannot decode the tag information from non-consecutive data bits, and so we cannot simply use the phase change modulation on the individual OFDM symbols in this context. To make the phase-change-based tag modulation still work for multi-stream signals, we have used a packet-level modulation, i.e., embedding one bit tag data on each packet. Since the same phase change operates on the whole data fields of a packet, increasing the packet length (the number of OFDM symbols) cannot increase the tag data rate anymore. For 802.11n and beyond, tag modulation using phase change reduces the deployment complexity, thereby allowing decoding with commodity WiFi NICs. To increase the backscatter throughput in MIMO, a tag can backscatter the same MIMO signal with multiple different frequency shifts and multiple MIMO WiFi NICs can be used to receive the backscattered signals at different channels. For instance, we can use two independent frequency shift modules (30MHz and 60MHz) for backscattering the original MIMO signal generated from Channel 1 (2.412GHz), and then receive the backscattered signals on Channel 7 (2.442GHz) and Channel 13 (2.472GHz), respectively. This would enable two bits of tag data for each multi-stream packet. The throughput can possibly be doubled, at the expense of adding tag hardware and higher requirement on synchronization between multiple receiving NICs.

8.2 Reconfigurable Tag

Programmable hardware has seen its growing use in the design of RFID tags [36–38]. Today's UHF RFID tags are typically write-capable and have a memory for information storage. This enables the reconfiguration of the tag identifier or the stored information, thereby remarkably extending the tag's functionality to support a variety of applications. Sensor tags can also be implemented with programmable microcontrollers or FPGA chips to work for diverse sensing applications [37]. State-of-the-art backscatter systems working with WiFi, such as Inter-Technology Backscatter [6] and HitchHike [8], have also used FPGA in their implementation of tag modulation. As discussed in Section 4, we use FPGA in the tag implementation for modulation as well, and extend its function for sensing applications. In particular, our phase-change modulation, frequency-shift signal generation, and control signal receiving modules are all implemented with FPGA digital circuits, so for the sensor drive circuits, as shown in Section 6. A concern here is the power consumption of the FPGAs. Recent studies, such as FM Backscatter [28] and HD Video Backscatter [39], adopt analog modulation circuits and Application-Specific Integrated Circuit (ASIC), so as to reduce the power consumption. Recent advances in low-power FPGAs will also improve the energy efficiency [40, 41].

8.3 Multiple Access Control

So far we have focused on the backscatter communication with a single MOXcatter tag. When multiple tags co-exist, e.g., in an indoor environment monitoring system that consists of different sensor tags communicating with a WiFi-based hub, a MAC protocol is needed to coordinate channel sharing among the tags. A possible solution is to use the control signals to notify the tags about the sending sequence. For example, when a tag receives the control signal containing a selected ID information, it compares the ID with its own; If they match, the tag can send data, and otherwise backs off. This way, all the tags can use the same channel for backscatter communication. Note that the receiver cannot receive the backscattered data from multiple tags simultaneously. This can be addressed by enabling the tags to backscatter from different channels and using multiple WiFi adapters to receive the backscattered data.

9 CONCLUSION

In this paper, we introduced MOXcatter, a backscatter communication system that explores WiFi signals with spatial multiplexing. We addressed the design challenges for spatial stream backscattering using off-the-shelf WiFi devices. We built a prototype hardware and showed that it can achieve throughput of up to 50Kbps for a single-stream and up to 1Kbps for a double-stream in 802.11n, with a range up to 14m for double streams. Though the preliminary implementation has tradeoffs, we believe that spatial stream backscatter communication has great potential in the new generation MIMO-enabled 802.11ac/ad/ax networks, and we will continue improving its performance towards high throughput and lower complexity, possibly with the Multi-user MIMO (MU-MIMO) technology introduced in such advanced WiFi standards as 802.11ac.

10 ACKNOWLEDGMENTS

We would like to thank the shepherd Shyamnath Gollakota and the anonymous reviewers for their valuable and insightful comments. This work was supported by a Canada Technology Demonstration Program (TDP) grant, a Canada NSERC Discovery Grant, and an NSERC E.W.R. Steacie Memorial Fellowship.

REFERENCES

[1] https://itstillworks.com/bluetooth-vs-wifi-power-consumption-17630.html
[2] S. Gollakota, M. S. Reynolds, J. R. Smith, and D. J. Wetherall. The Emergence of RF-Powered Computing. *Computer*, 47(1): 32-39, 2014.
[3] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall. Wi-Fi Backscatter: Internet Connectivity for RF-Powered Devices. In *ACM SIGCOMM*, 2014.
[4] D. Bharadia, K. Joshi, M. Kotaru, and S. Katti. BackFi: High Throughput WiFi Backscatter. In *ACM SIGCOMM*, 2015.
[5] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith. Passive Wi-Fi: Bringing Low Power to Wi-Fi Transmissions. In *USENIX NSDI*, 2016.
[6] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. R. Smith. Inter-Technology Backscatter: Towards Internet Connectivity for Implanted Devices. In *ACM SIGCOMM*, 2016.
[7] P. Zhang, M. Rostami, P. Hu, and D. Ganesan. Enabling Practical Backscatter Communication for On-body Sensors. In *ACM SIGCOMM*, 2016.
[8] P. Zhang, D. Bharadia, K. Joshi, and S. Katti. HitchHike: Practical Backscatter Using Commodity WiFi. In *ACM SenSys*, 2016.
[9] O. Abari, D. Vasisht, D. Katabi, and A. Chandrakasan. Caraoke: An E-Toll Transponder Network for Smart Cities. In *ACM SIGCOMM*, 2015.
[10] J. Gummeson, P. Zhang, and D. Ganesan. Flit: A Bulk Transmission Protocol for RFID-Scale Sensors. In *ACM MobiSys*, 2012.
[11] H. Hassanieh, J. Wang, D. Katabi, and T. Kohno. Securing RFIDs by Randomizing the Modulation and Channel. In *USENIX NSDI*, 2015.
[12] P. Hu, P. Zhang, and D. Ganesan. Leveraging Interleaved Signal Edges for Concurrent Backscatter. In *ACM Workshop on Hot Topics in Wireless*, 2014.
[13] P. Hu, P. Zhang, and D. Ganesan. Laissez-faire: Fully Asymmetric Backscatter Communication. In *ACM SIGCOMM*, 2015.
[14] P. Hu, P. Zhang, M. Rostami, and D. Ganesan. Braidio: An Integrated Active-Passive Radio for Mobile Devices with Asymmetric Energy Budgets. In *ACM SIGCOMM*, 2016.
[15] B. Kellogg, V. Talla, and S. Gollakota. Bringing Gesture Recognition to All Devices. In *USENIX NSDI*, 2014.
[16] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient Backscatter: Wireless Communication Out of Thin Air. In *ACM SIGCOMM*, 2013.
[17] V. Liu, V. Talla, and S. Gollakota. Enabling Instantaneous Feedback with Full-Duplex Backscatter. In *ACM MobiCom*, 2014.
[18] A. N. Parks, A. Liu, S. Gollakota, and J. R. Smith. Turbocharging Ambient Backscatter Communication. In *ACM SIGCOMM*, 2014.
[19] V. Talla, B. Kellogg, B. Ransford, S. Naderiparizi, S. Gollakota, and J. R. Smith. Powering the Next Billion Devices with Wi-Fi. In *ACM CoNEXT*, 2015.
[20] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus. RF-Compass: Robot Object Manipulation Using RFID. In *ACM MobiCom*, 2013.
[21] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and Reliable Low-Power Backscatter Networks. In *ACM SIGCOMM*, 2012.
[22] J. Wang and D. Katabi. Dude, Where's My Card?: RFID Positioning That Works with Multipath and Non-Line of Sight. In *ACM SIGCOMM*, 2013.
[23] J. Wang, D. Vasisht, and D. Katabi. Rf-idraw: Virtual Touch Screen in the Air Using RF Signals. In *ACM SIGCOMM*, 2014.
[24] P. Zhang and D. Ganesan. Enabling Bit-by-Bit Backscatter Communication in Severe Energy Harvesting Environments. In *USENIX NSDI*, 2014.
[25] P. Zhang, D. Ganesan, and B. Lu. Quarkos: Pushing the Operating Limits of Micro-Powered Sensors. In *USENIX Workshop on Hot Topics in Operating Systems*, 2014.
[26] P. Zhang, J. Gummeson, and D. Ganesan. Blink: A High Throughput Link Layer for Backscatter Communication. In *ACM MobiSys*, 2012.
[27] P. Zhang, P. Hu, V. Pasikanti, and D. Ganesan. Ekhonet: High Speed Ultra Low-Power Backscatter for Next Generation Sensors. In *ACM MobiCom*, 2014.
[28] A. Wang, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota. FM Backscatter: Enabling Connected Cities and Smart Fabrics. In *USENIX NSDI*, 2017.
[29] V. Talla, B. Kellogg, S. Gollakota, and J. R. Smith. Battery-free Cellphone. In *ACM UbiComp*, 2017.
[30] P. Zhang, C. Josephson, D. Bharadia, and S. Katti. FreeRider: Backscatter Communication Using Commodity Radios. In *ACM CoNEXT*, 2017.
[31] IEEE Std 802.11-1999. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.
[32] IEEE Std 802.11a-1999. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHZ Band.
[33] IEEE Std 802.11g-2003. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band.
[34] IEEE Std 802.11n-2009. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 5: Enhancements for Higher Throughput.
[35] http://www.tamos.com/products/commwifi/
[36] A. K. Jones, R. Hoare, S. Dontharaju, S. Tung, R. Sprang, J. Fazekas, J. T. Cain, and M. H. Mickle. An Automated, FPGA-based Reconfigurable, Low-power RFID Tag. *ELSEVIER Microprocessors and Microsystems*, 31(2): 116-134, 2007.
[37] A. P. Sample, D. J. Yeager, P. S. Powledge, A. V. Mamishev, and J. R. Smith. Design of An RFID-based Battery-free Programmable Sensing Platform. *IEEE Trans. Instrumentation and Measurement*, 57(11): 2608-2615, 2008.
[38] D. J. Yeager, A. P. Sample, J. R. Smith, and J. R. Smith. Wisp: A Passively Powered UHF RFID Tag with Sensing and Computation. *RFID handbook: Applications, technology, security, and privacy*, (2008): 261-278, 2008.
[39] S. Naderiparizi, M. Hessar, V. Talla, S. Gollakota, and J. R. Smith. Towards Battery-Free HD Video Streaming. In *USENIX NSDI*, 2018.
[40] https://www.xilinx.com/products/technology/power.html
[41] http://www.actel.com/FPGA/handheld/?p=sn
[42] M. Wong, J. M. Gilbert, and C. H. Barratt. Wireless LAN using RSSI and BER parameters for transmission rate adaptation. *US patent* 7,369,510, 2008.
[43] I. Pefkianakis, Y. Hu, S. H. Y. Wong, H. Yang, and S. Lu. MIMO Rate Adaptation in 802.11n Wireless Networks. In *ACM MobiCom*, 2010.