

Wi-Fi Backscatter: Internet Connectivity for RF-Powered Devices

Bryce Kellogg, Aaron Parks, Shyamnath Gollakota, Joshua R. Smith, and David Wetherall
University of Washington
{kellogg, anparks, gshyam, jrjsrs, djw}@uw.edu

Abstract— RF-powered computers are small devices that compute and communicate using only the power that they harvest from RF signals. While existing technologies have harvested power from ambient RF sources (e.g., TV broadcasts), they require a dedicated gateway (like an RFID reader) for Internet connectivity.

We present *Wi-Fi Backscatter*, a novel communication system that bridges RF-powered devices with the Internet. Specifically, we show that it is possible to reuse existing Wi-Fi infrastructure to provide Internet connectivity to RF-powered devices. To show Wi-Fi Backscatter’s feasibility, we build a hardware prototype and demonstrate the first communication link between an RF-powered device and commodity Wi-Fi devices. We use off-the-shelf Wi-Fi devices including Intel Wi-Fi cards, Linksys Routers, and our organization’s Wi-Fi infrastructure, and achieve communication rates of up to 1 kbps and ranges of up to 2.1 meters. We believe that this new capability can pave the way for the rapid deployment and adoption of RF-powered devices and achieve ubiquitous connectivity via nearby mobile devices that are Wi-Fi enabled.

CATEGORIES AND SUBJECT DESCRIPTORS

C.2.1 [Network Architecture and Design]: Wireless communication

KEYWORDS

Backscatter; Internet of Things; Energy harvesting; Wireless

1. INTRODUCTION

It is increasingly becoming feasible to perform low-end computing, sensing, and more recently communication [17] by harvesting power from ambient RF signals including TV, cellular, and Wi-Fi transmissions [26, 27, 22]. This technology is intriguing because it may be embedded inexpensively into everyday objects to help realize the pervasive vision of the “Internet of Things” [9]: objects that operate and communicate with each other when they are nearby, all without the need to ever plug them in or maintain batteries.

Existing technologies, however, lack the central component in this vision of an RF-powered Internet of Things: an ability to connect RF-powered devices to the Internet. Using conventional radio communication to achieve this connectivity is challenging, since it consumes orders of magnitude more power than is available in ambient RF [23]. Communication techniques like ambient backscatter [17] enable these devices to communicate with each other; but

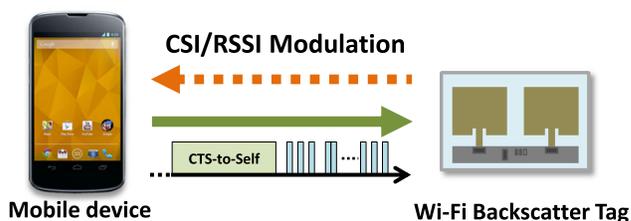


Figure 1—Wi-Fi Backscatter at a high level. The figure shows a Wi-Fi Backscatter tag (an RF-powered device) communicating with a Wi-Fi enabled mobile device. On the uplink, the Wi-Fi Backscatter tag conveys information by modulating the CSI and RSSI measurements received at the Wi-Fi device. On the downlink, the Wi-Fi device conveys information by encoding bits in the presence or silence of Wi-Fi packets. The Wi-Fi Backscatter tag uses its low-power Wi-Fi packet detector circuit to decode this information.

they create an isolated network disconnected from the Internet. A naïve solution is to develop and deploy special-purpose powered infrastructure devices, along the lines of RFID readers, that can connect the two networks. But deploying new support infrastructure adds cost and, more importantly, diminishes the key value-proposition of RF-powered systems — an ability to operate without dedicated power infrastructure.

In this paper, we ask if it is possible to reuse existing infrastructure to connect these devices to the Internet. Specifically, we seek to design RF-powered devices that communicate directly with commodity Wi-Fi devices. A positive answer would pave the way for a rapid and simple deployment of the RF-powered Internet of Things by letting these devices connect to existing mobile phones and Wi-Fi APs. It would also expand the functionality of Wi-Fi networks in a new direction: from providing connectivity to existing Wi-Fi clients to a whole new class of battery-free devices.

Achieving this capability, however, is challenging since conventional low-power Wi-Fi transceivers require much more power than is available from ambient RF signals. Thus, it is not feasible for RF-powered devices to literally speak the Wi-Fi protocol. Conversely, since existing Wi-Fi devices are specifically designed to receive Wi-Fi signals, it is unclear how they would decode other kinds of signals from RF-powered devices.

We introduce *Wi-Fi Backscatter*, a novel communication system in which RF-powered “things” can communicate with off-the-shelf Wi-Fi devices. At a high level, a Wi-Fi Backscatter tag communicates with a Wi-Fi device by modulating its Wi-Fi channel. Since channel information including CSI and RSSI is widely available on commodity Wi-Fi devices, the Wi-Fi receiver can extract the modulated information by measuring the changes in its channel. To understand this in more detail, consider the Wi-Fi Backscatter tag and Wi-Fi enabled mobile device in Fig. 1. The Wi-Fi Backscatter tag communicates by modulating the Wi-Fi channel as seen by the mobile device. Specifically, it conveys a ‘1’ and a ‘0’ bit by either reflecting or absorbing the Wi-Fi packets received by the mobile device. The reflected signals change the per-packet CSI and RSSI

measurements, which the mobile device uses to decode messages from the Wi-Fi Backscatter tag.

We also enable downlink communication to the Wi-Fi Backscatter tag. While the mobile device can send Wi-Fi packets directly to the tag, the latter cannot decode Wi-Fi transmissions. Instead, Wi-Fi Backscatter relies on low-power circuit designs that can detect energy corresponding to a Wi-Fi packet (see §4.2). Specifically, as shown in the figure, the mobile device sends a pattern of short Wi-Fi packets — the presence (absence) of the short Wi-Fi packet encodes the ‘1’ (‘0’) bit. The Wi-Fi Backscatter tag decodes this information by using our low-power energy detector to differentiate between the presence and absence of Wi-Fi packets. In §4, we develop the above ideas further and design Wi-Fi Backscatter to work in the presence of multiple Wi-Fi devices in the network.

To show the feasibility of our designs, we build prototype devices that are optimized to backscatter and harvest Wi-Fi signals at frequencies spanning the entire 2.4 GHz Wi-Fi band. We evaluate our prototype in various scenarios with commodity Wi-Fi devices including Intel Wi-Fi cards and the Wi-Fi infrastructure in our organization. We use the RSSI information that is widely available on commodity Wi-Fi devices. We also use the Intel CSI toolkit [13] to extract the finer grained CSI information. Our results show the following.

- The Wi-Fi devices can reliably decode information on the uplink at distances of up to 65 cm and 30 cm using CSI and RSSI information respectively. This is achieved at bit rates ranging from 100 bps to 1 kbps. The uplink range can be increased to more than 2.1 meters by performing coding at the Wi-Fi device.
- The uplink can operate using only the ambient Wi-Fi transmissions in the network. Specifically, the Wi-Fi device can use RSSI information extracted from all the packets transmitted by the AP in our organization to achieve uplink bit rates between 100 bps and 200 bps, depending on the network load.
- The prototype can detect Wi-Fi packets as short as 50 μ s at distances of up to 2.2 meters; this translates to a bit rate of 20 kbps on the downlink. The downlink range can be further extended to about 3 meters by reducing the bit rate to 5 kbps.

Contributions: We make the following contributions:

- We introduce a novel communication system that connects RF-powered devices (Wi-Fi Backscatter tags) to the Internet via existing Wi-Fi infrastructure.
- We design an uplink communication channel that allows Wi-Fi Backscatter tags to convey information to Wi-Fi devices by modulating the Wi-Fi channel information including CSI and RSSI.
- We design a downlink communication channel where Wi-Fi devices encodes information in the presence or absence of Wi-Fi packets; the Wi-Fi Backscatter tags use low-power circuits to detect Wi-Fi packets and decode this information.
- Finally, we build a hardware prototype for Wi-Fi Backscatter and demonstrate the first communication link between a Wi-Fi Backscatter tag and commodity Wi-Fi devices.

While the bit rates demonstrated in this paper are not high, we believe that the ability to communicate at even low rates between RF-powered devices and existing Wi-Fi infrastructure is a significant capability that would be critical for the commercial adoption of an RF-powered Internet of Things.

2. WI-FI BACKSCATTER OVERVIEW

Wi-Fi Backscatter is a novel communication system that enables RF-powered devices to communicate directly with commodity Wi-Fi devices. As shown in Fig. 2, Wi-Fi Backscatter has three main

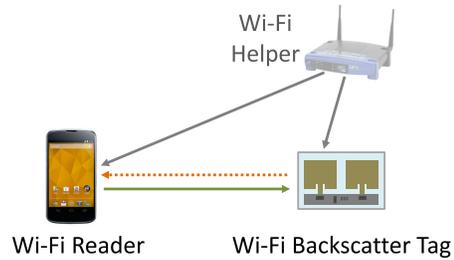


Figure 2—Wi-Fi Backscatter Overview. It has three actors: a Wi-Fi reader, a Wi-Fi helper, and a RF-powered device (Wi-Fi Backscatter tag). The reader and helper can be any commodity Wi-Fi device, including routers and mobile devices. The Wi-Fi reader leverages the channel information from the Wi-Fi helper’s packets to decode the transmissions from the tag.

actors: a Wi-Fi reader, a Wi-Fi helper, and an RF-powered Wi-Fi Backscatter tag. In principle, the reader and helper can be any commodity Wi-Fi device, including routers and mobile devices. However, for ease of exposition, we consider a Wi-Fi client, such as a mobile phone, to be the reader and a Wi-Fi AP to be the helper device. We envision the RF-powered devices to be wireless sensor nodes that are embedded in everyday objects. The Wi-Fi Backscatter tags do not require any batteries and can harvest energy from ambient RF signals including TV, cellular, and Wi-Fi. We also note that Wi-Fi Backscatter can also be used to provide Internet connectivity to battery-free devices with other power harvesting sources including solar and mechanical energy.

Wi-Fi Backscatter’s communication has two main components: an uplink from the Wi-Fi Backscatter tag to the Wi-Fi reader and a downlink from the Wi-Fi reader to the Wi-Fi Backscatter tag. Wi-Fi Backscatter follows a request-response model, similar to RFID systems. Specifically, the Wi-Fi reader asks the Wi-Fi Backscatter tag for information on the downlink and receives a response on the uplink. In the presence of multiple Wi-Fi Backscatter tags in the vicinity, the interrogator can use protocols similar to EPC Gen-2 [2] to identify these devices and then query each of them individually. In this paper, however, we focus on establishing a communication link with a single Wi-Fi Backscatter tag.

At a high level, Wi-Fi Backscatter imitates RFID communication using a Wi-Fi device. Specifically, the Wi-Fi Backscatter tag communicates by modulating the transmissions from the Wi-Fi helper. The Wi-Fi reader decodes these transmissions by using the channel changes that are created on the received Wi-Fi packets. In principle, Wi-Fi Backscatter can have multiple Wi-Fi helper devices (e.g., multiple APs or other Wi-Fi client); in this paper, however, we focus on using a single Wi-Fi helper device. For the downlink, the Wi-Fi reader transmits directly to the Wi-Fi Backscatter tag by encoding information in short Wi-Fi packets interspersed by silence. The tag decodes these transmissions using a low-power circuit design that can detect energy from Wi-Fi packets.

In the rest of this paper, we describe the design of the Wi-Fi Backscatter uplink in §3 and the downlink in §4. We then generalize our description to work in the presence of multiple Wi-Fi devices in §5. Finally, we evaluate our prototype in various scenarios.

3. WI-FI BACKSCATTER UPLINK

Wi-Fi Backscatter enables communication on the uplink from an RF-powered device to the Wi-Fi reader. This is challenging however because it is not feasible for these devices to speak the Wi-Fi protocol. Instead we transmit data by modulating the Wi-Fi channel as seen in the Wi-Fi reader channel state information. In this section we describe the Wi-Fi Backscatter uplink in more detail.

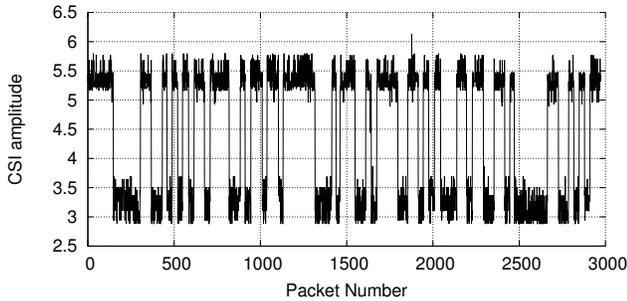


Figure 3—Raw CSI measurements for a single Wi-Fi sub-channel in the presence of the Wi-Fi Backscatter tag 5 centimeters away. The plot clearly shows a binary modulation on top of the CSI measurements.

3.1 Modulating Wi-Fi Channel at the tag

The Wi-Fi Backscatter tag conveys information by modulating the Wi-Fi channel. To do this, it uses a low power switch that allows the antenna impedance to be modulated by a transmitted bit stream. The antenna’s impedance affects the amount of signal that is reflected by the tag. By modulating this impedance, the tag can convey ‘1’ and ‘0’ bits. These switches consume less than $1 \mu\text{W}$ of power, which is negligible. Our prototype design optimizes the choice of the switch to operate well at frequencies spanning the entire 2.4 GHz Wi-Fi band.

The Wi-Fi Backscatter tag can in principle use a variety of antennas including monopoles and dipoles. For our prototype, we designed a special-purpose antenna for the Wi-Fi Backscatter tag that is capable of both impedance modulation as well as Wi-Fi energy harvesting. Specifically, we design a microstrip “patch” antenna, as the gain of this antenna can be relatively high with respect to its size. Our antenna design takes into consideration a key factor of system performance: the extent of radar cross section changes that the antenna is capable of achieving. The radar cross-section refers to the amount of incident Wi-Fi signals that can be reflected by an antenna; the contrast between the radar cross-section when the tag is reflecting versus not reflecting will determine the impact of the tag on a nearby Wi-Fi receiver. To improve the radar cross-section, we design an array of microstrip patches, each of which resonates at 2.4 GHz, but which act together to collaboratively scatter incident RF and therefore produce a larger change in the radar cross-section. Wi-Fi antenna design is a broad topic and one can use other antenna topologies that can significantly reduce the size of the antenna [18]. This, however, is not in the scope of this paper.

Finally, the minimum period with which our tag changes its impedance is larger than the duration of a Wi-Fi packet. That is, the duration of each transmitted bit is greater than the time it takes to transmit a few Wi-Fi packets. This ensures that the act of modulating does not change the channel within every Wi-Fi packet, thus allowing nearby Wi-Fi communication to proceed with minimal interference. We also note that the Wi-Fi Backscatter tag modulates the Wi-Fi channel only when queried by the reader. Further, the tag can adapt the modulation rate by increasing the duration of each bit, enabling our tag to adapt the resulting data rate to the Wi-Fi network traffic conditions (see §5).

3.2 Decoding algorithm at the Wi-Fi reader

Next, we describe how the Wi-Fi reader extracts the modulated Wi-Fi channel corresponding to the packets from the helper. To understand how the reflections at a Wi-Fi Backscatter tag affect the Wi-Fi channel, we run the following experiment. We place an Intel Wi-Fi Link 5300 card, which acts as a Wi-Fi reader, next to a

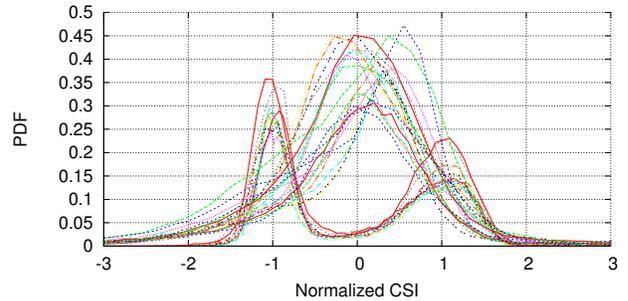


Figure 4—PDF of normalized channel values for the 30 Wi-Fi sub-channels. The plot shows diversity between sub-channels.

Wi-Fi Backscatter tag. We configure the tag to modulate an alternating sequence of zero and one bits. The Wi-Fi reader downloads a 1 GB media file from a Linksys WRT54GL AP, which acts as a Wi-Fi helper. The helper is placed 5 meters away from the Wi-Fi Backscatter tag. Fig. 3 plots the CSI in Wi-Fi sub-channel 19 as a function of the packet number. The plot clearly shows a binary modulation on top of the captured channel measurements.

Our decoding algorithm extracts this modulated information in three main steps: (1) Signal conditioning to remove the variations in the channel measurements due to mobility in the environment, (2) Leveraging frequency diversity across the Wi-Fi sub-channels, and (3) Decoding the backscattered bits from the channel information.

(1) Signal Conditioning. The goal of this step is two-fold: 1) remove the natural temporal variations in the channel measurements due to mobility in the environment, and 2) normalize the channel measurements to map to -1 and +1 values. In particular, to remove the temporal variations, we subtract a moving average from the channel measurements at the Wi-Fi reader; the moving average is computed over a duration of 400 ms in our experiments. The above operation creates zero-mean channel measurements without the temporal variations. Next, we normalize these measurements such that the channel corresponding to a one bit from the Wi-Fi Backscatter tag maps to a +1 and that corresponding to a zero bit maps to a -1. Since we do not know the actual bits transmitted, we instead perform this normalization by computing the absolute values of the zero-mean channel measurements and then taking their average. We then divide the zero-mean channel measurements with this computed average to get the normalized channel values.

(2) Exploiting Wi-Fi Frequency/Spatial Diversity. Wi-Fi transmissions span a bandwidth of 20 MHz. At such a wide bandwidth, it is well known that the signals experience frequency diversity where the amplitude and the phase of the channel measurements vary significantly with the Wi-Fi OFDM subcarriers. Similarly, multipath issues result in similar variations across antennas. Thus, one would expect that some of the Wi-Fi subcarriers and antennas experience a stronger effect due to the reflections from the Wi-Fi Backscatter tag. To empirically check this effect, Fig. 4 plots the probability density functions (PDF) of the normalized channel values for each adjacent pair of the 60 Wi-Fi subcarriers (resulting in 30 “sub-channels”) available from the Intel card. The PDF is computed over channel measurements taken across 42,000 Wi-Fi packets. Fig. 4 shows three main points: First, for about 30 percent of the Wi-Fi sub-channels, we see two Gaussian signals centered at +1 and -1; these represent the two-reflection states at the tag. Secondly, the variance in the channel measurements, i.e., noise, changes significantly with the sub-channel being used. Finally, some of the sub-channels do not see two distinct Gaussian signals. This corresponds to frequencies where the effect of backscatter is very weak, due to

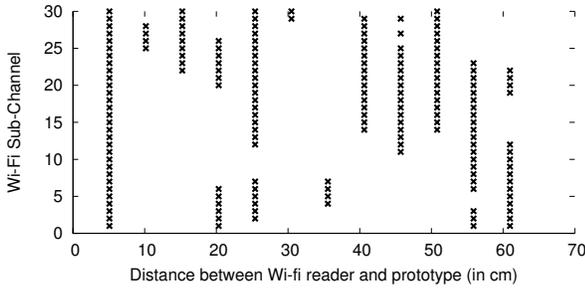


Figure 5—Wi-Fi sub-channels with BER $< 10^{-2}$ at various distances. For each Wi-Fi sub-channel, the figure shows the experiments where decoding using only that sub-channel achieves a bit error rate less than 10^{-2} . The plot shows that the set of good sub-channels varies significantly with the position (and therefore multipath profile) of the Wi-Fi Backscatter tag.

multipath effects. Thus, we observe significant frequency diversity within the 30 Wi-Fi sub-channels even when the Wi-Fi Backscatter tag is placed next to the Wi-Fi reader.

We exploit this frequency diversity to increase the reliability of Wi-Fi Backscatter’s uplink communication. Specifically, the Wi-Fi reader performs the following two steps:

Step 1: Identify the good Wi-Fi sub-channels/antennas. Ideally, if the combination of a set of Wi-Fi sub-channels and antennas that have a strong signal from the Wi-Fi Backscatter tag is consistent across locations, then we could decode the information from the tag using only that combination. The challenge however is that the set of “good” sub-channels and antennas varies significantly with the position of the tag. To empirically see this, we place the Wi-Fi Backscatter tag at different distances from the Wi-Fi reader and measure the channel information on all the Wi-Fi sub-channels. Fig. 5 shows the sub-channels where the Wi-Fi reader could decode the modulated information with a BER of less than 10^{-2} at a given distance. The figure shows that, in general, there are no Wi-Fi sub-channels that are consistently good. Thus, the Wi-Fi reader should independently extract the good sub-channel information for each uplink transmission.

To do this, the Wi-Fi reader uses correlation with a known preamble from the tag; the preamble is used by the tag at the beginning of each of its message transmission. Specifically, the Wi-Fi reader correlates with the preamble along every sub-channel (treating multiple antennas as additional sub-channels), while waiting for an incoming transmission from the tag. When a transmission arrives (which is identified by a peak in the correlation), the Wi-Fi reader sorts the sub-channels based on the correlation value. In particular, the sub-channels that correlate well with the preamble have a better signal from the Wi-Fi Backscatter tag than those that correlate poorly. The Wi-Fi reader picks the top ten “good” sub-channels based on this correlation method.

Step 2: Combining information across the good Wi-Fi sub-channels. A naive approach to do this is to simply add up the information across all the sub-channels. This is, however, not optimal because the noise variance may vary even across the good sub-channels. Instead the Wi-Fi reader combines the information across the sub-channels by computing a weighted average where sub-channels with low noise variance are given a higher weight, while those with higher noise variance are given a lower weight.

More formally, say CSI_i is the normalized CSI computed on the i th good Wi-Fi sub-channel. The Wi-Fi reader performs a linear combination of the normalized CSI across these sub-channels by weighting them with the noise variance. Specifically, the reader

computes the following summation:

$$CSI_{weighted} = \sum_{i=1}^G \frac{CSI_i}{\sigma_i^2}$$

where G is the total number of good sub-channels, and σ_i^2 is the noise variance in the i th sub-channel. Effectively, the above equation gives a larger weight to Wi-Fi sub-channels where the noise variance is low (and hence a higher confidence). The above computation is similar to maximum ratio combining techniques [7] used in traditional communication that are known to be optimal for Gaussian noise. In scenarios where the Wi-Fi reader has multiple antennas, the above computation can be performed for each antenna and the summation can be taken across all the antennas.

(3) Decoding bits from the Wi-Fi Backscatter tag. To do this, the Wi-Fi reader can use a simple thresholding mechanism on $CSI_{weighted}$. Specifically, if $CSI_{weighted}$ is greater than zero, the receiver outputs a ‘1’ and a ‘0’ otherwise. We note the following:

- The CSI information provided by the off-the-shelf Wi-Fi devices is inherently noisy. To account for these noisy measurements, we add redundancy to our transmissions. Specifically, each bit transmitted by the tag corresponds to multiple channel measurements (i.e., received Wi-Fi packets) at the receiver. The Wi-Fi reader repeats the above procedure for all these channel measurements and uses a simple majority vote to compute the transmitted bits.
- Since the Wi-Fi medium is shared and bursty in nature, it is unlikely that every bit transmitted by the tag sees the same number of Wi-Fi packets (and the corresponding channel measurements). To account for this, we use the timestamp that is in every Wi-Fi packet header to accurately group Wi-Fi packets belonging to the same bit transmission. We then perform majority voting over the corresponding channel measurements.
- Finally, the Intel cards used in our experiments report spurious changes in the CSI once every so often. We see this behavior even in a static network with no mobility. To account for this spurious behavior, we use a hysteresis mechanism. Specifically, we use two thresholds, $Thresh_0$ and $Thresh_1$, corresponding to the 0 and 1 bits. The receiver outputs a one (zero) bit only when the received channel value is greater (smaller) than $Thresh_1$ ($Thresh_0$). We set the threshold values to be $\mu \pm \sigma^2$, where μ and σ are the mean and standard deviation of $CSI_{weighted}$ computed across packets. The above heuristic works effectively in our experiments.

3.3 Decoding Using RSSI

While the 802.11n Wi-Fi specification requires per-subchannel channel state information to be made available on newer Wi-Fi chipsets, most existing chipsets only provide the RSSI information [5]. RSSI is a single metric that provides a measure of the cumulative Wi-Fi signal strength across all the sub-channels. In this section, we describe how the Wi-Fi reader can decode the reflected information from the Wi-Fi Backscatter tag using only RSSI.

To do this, we employ a decoding algorithm similar to that described earlier. Specifically, we perform signal conditioning, hysteresis, and thresholding to decode the information exactly as in the algorithm from §3.2. We note that depending on the Wi-Fi chipset, the RSSI information available is either a single value per packet or an RSSI value per antenna in the case of MIMO receivers. In scenarios with multiple RSSI channels (e.g., multiple antennas), we select the best RSSI channel using the maximum correlation mechanism. In particular, the receiver correlates with the packet preamble and picks the RSSI channel that has the maximum correlation value. Finally, we note that since RSSI is a single value that repre-

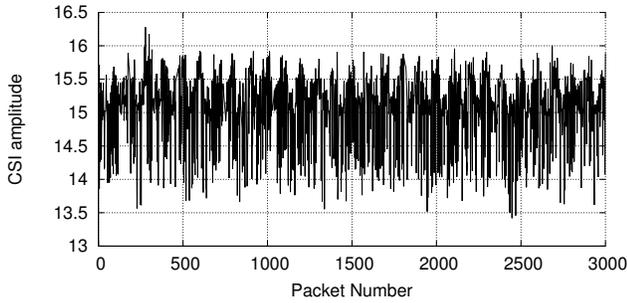


Figure 6—Raw CSI measurements for a single Wi-Fi sub-channel in the presence of the Wi-Fi Backscatter tag one meter away. The figure shows that at larger ranges, there are no longer two distinct levels in the CSI measurements. Thus, we need to design a different decoding mechanism to achieve higher ranges.

sents all Wi-Fi sub-channels and the RSSI bit resolution is limited, the BER performance is better with CSI information than RSSI.

3.4 Increasing Uplink Communication Range

The algorithm described so far assumes that the reflections from the Wi-Fi Backscatter tag create a distinctive difference in the channel values between a one and a zero bit. While such an algorithm is effective at small distances from the Wi-Fi reader (in our experiments up to 65 centimeters), it breaks down at larger distances. To see this, consider the channel measurements in Fig. 6, at a distance of two meters between the Wi-Fi Backscatter tag and the Wi-Fi reader. The figure shows that there are no two distinct levels in the channel measurements, which is in contrast to Fig. 3.

Wi-Fi Backscatter uses coding to increase its uplink communication range. Specifically, the tag transmits two orthogonal codes of length L each, to represent the one and the zero bits. The Wi-Fi reader correlates the channel measurements with the two codes and outputs the bit corresponding to the larger correlation value.

- Wi-Fi Backscatter repeats the above correlation operation on all the frequency sub-channels and picks the Wi-Fi sub-channels that provide the maximum correlation peaks.
- The communication range of the system can be increased by increasing the code length, L . This is because correlation with a L bit long code provides an increase in the SNR that is proportional to L . Our evaluation shows that with a correlation length of 20 bits, the communication range can be increased to 1.6 meters. The uplink communication range can be further increased to 2.1 meters by increasing the correlation length to 150 bits..
- Since the tag still only transmits bits (now the bit duration expanded by L) and does not perform any decoding operations, the power consumption of the tag does not increase. The Wi-Fi reader on the other hand is a powered device and can perform the above correlation operations.

4. WI-FI BACKSCATTER DOWNLINK

Next, we describe how Wi-Fi Backscatter enables communication on the downlink from the Wi-Fi reader to a Wi-Fi Backscatter tag. The challenge in achieving this is that, on one hand, the reader can only transmit Wi-Fi packets; on the other hand, a Wi-Fi Backscatter tag cannot decode Wi-Fi transmissions. Instead, we design a novel circuit for the tag that can detect the energy during a Wi-Fi packet from a nearby transmitter. We then have the Wi-Fi reader encode information in the presence and absence of Wi-Fi packets. In the rest of this section, we describe the encoding mechanism at the Wi-Fi reader and the receiver design at the tag.

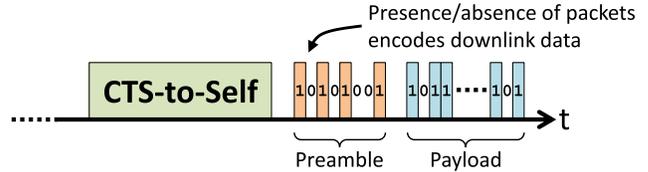


Figure 7—Encoding information on the downlink. the reader encodes a ‘1’ bit with presence of a Wi-Fi packet and a ‘0’ bit with silence (i.e., the absence of Wi-Fi packets). The duration of the silence period is set to be equal to that of the Wi-Fi packet. To enforce other Wi-Fi devices in the vicinity to not transmit during the silence periods, the Wi-Fi reader transmits a *CTS_to_SELF* packet before transmitting the message. The message consists of both the preamble bits and the payload bits (including the CRC).

4.1 Encoding at the Wi-Fi reader

The Wi-Fi reader encodes information using the presence or absence of a Wi-Fi packet. Specifically, as shown in Fig. 7, the reader encodes a ‘1’ bit with presence of a Wi-Fi packet and a ‘0’ bit with silence (i.e., the absence of Wi-Fi packets). The duration of the silence period is set to be equal to that of the Wi-Fi packet. To force other Wi-Fi devices in the vicinity to not transmit during the silence periods, the Wi-Fi reader transmits a *CTS_to_SELF* packet [10] before transmitting the encoded bits. *CTS_to_SELF* is a Wi-Fi message that forces 802.11-compliant devices to refrain for a specified time period. Wi-Fi Backscatter leverages this message to reserve the medium for the duration of its transmission to ensure that other Wi-Fi devices, unaware of the Wi-Fi Backscatter protocol, do not sense the medium as idle and transmit during the silence periods. We note the following key points:

- The current 802.11 standard only allows Wi-Fi devices to reserve the channel for up to a duration of 32 ms using the *CTS_to_SELF* packet. Thus, to maximize the number of bits transmitted within this duration, the Wi-Fi reader reduces the size of each Wi-Fi packet and the corresponding silence period. The smallest packet size possible on a Wi-Fi device is about $40 \mu\text{s}$ at a bit rate of 54 Mbps. Our receiver design in §4.2 can reliably detect Wi-Fi packets as small as $50 \mu\text{s}$ to a distance of more than two meters. Thus, the Wi-Fi reader can transmit a 64-bit payload message with a 16-bit preamble in 4.0 ms. We can transmit more bits by splitting them across multiple *CTS_to_SELF* packets.
- Since we assume a query-response model in this paper, if the Wi-Fi Backscatter tag does not respond to the Wi-Fi reader’s query, the reader re-transmits its packet until it gets a response. More generally, the Wi-Fi Backscatter tag could also transmit ACK packets back to the Wi-Fi reader using the uplink communication in §3. The Wi-Fi Backscatter tag can reduce the overhead of the ACK packet by dropping the preamble and the address fields, and transmitting a single bit message.
- One may be worried that since Wi-Fi Backscatter’s downlink messages occupy a considerable amount of time, they could affect the efficiency of Wi-Fi networks. While this certainly is a factor to be considered, there is growing evidence that Wi-Fi networks in typical home and office scenarios are underutilized [24]. Thus, we believe that the spare Wi-Fi bandwidth can be used to communicate with Wi-Fi Backscatter devices.

4.2 Wi-Fi Backscatter tag receiver design

The goal of our receiver is to differentiate between the presence and absence of a Wi-Fi packet and decode the bits transmitted by the Wi-Fi reader. At a high level, we design a low-power Wi-Fi energy detection circuit that leverages the RF harvesting capabilities of our device. Traditional energy detection approaches compute the

average energy in the received signal and use a highly sensitive receiver to detect the presence of energy on the wireless medium. This approach however is not suitable in our scenario since the receiver is low power in nature and hence has a very low sensitivity. Further, Wi-Fi transmissions are modulated using OFDM, which is known to have a high peak to average ratio [20]. Said differently, the average energy in the Wi-Fi signal is small, with occasional peaks spread out during the transmission.

Thus, the Wi-Fi Backscatter tag leverages a specially designed RF energy detector based on peak detection to decode information from the Wi-Fi Backscatter reader. As shown in Fig. 8, our receiver circuit has four main components: an envelope detector, a peak finder, a set-threshold circuit and a comparator. The role of the envelope detector circuit is to remove the carrier frequency (2.4 GHz) of the Wi-Fi transmissions. This is a standard circuit design similar to that used in RFID systems. We however tune the circuit elements to be optimal over the whole 2.4 GHz Wi-Fi frequency ranges.

The peak detector circuit captures and holds the peak amplitude of the received signal. It uses a diode, an operational amplifier, and a capacitor that can store the peak amplitudes as its charge. In order to adapt to time-varying channel conditions, it however does not hold this peak value indefinitely; the resistor network that is part of the set-threshold circuit allows the charge on the capacitor to slowly dissipate, effectively “resetting” the peak detector over some relatively long time interval. The output of this peak-detection circuit is halved to produce the actual threshold; this is performed by the capacitor element in the set-threshold circuit. Finally, the comparator takes two inputs: the threshold value and the received signal, and outputs a one bit whenever the received signal is greater than the threshold value and a zero bit otherwise.

We note that the receiver circuit described above can detect packets as small as $50 \mu\text{s}$. Thus, it can differentiate Wi-Fi packet lengths up to that resolution. Specifically, since longer packets can be intuitively thought of as multiple small packets sent back-to-back without any gap, the Wi-Fi Backscatter tag outputs a continuous sequence of ones corresponding to each long packet. By counting the number of ones, Wi-Fi Backscatter can resolve the length of a Wi-Fi packet to a resolution of $50 \mu\text{s}$.

Further, the above circuit requires only a very small amount of power to operate (around $1 \mu\text{W}$), and can therefore be left ON at all times. However, the microcontroller requires a relatively large amount of power (several hundred μW) in its active mode. To reduce overall power consumption, the Wi-Fi Backscatter tag keeps the microcontroller in a sleep state as much as possible by operating under two main modes:

- *Preamble detection mode.* The receiver spends most of its time in this mode detecting preambles at the beginning of potential reader transmissions. To reduce the power consumption of doing this, we leverage that there is no information in between the transitions of bits output by our receiver circuit. Thus, we keep the microcontroller asleep until a new transition occurs at the comparator’s output. We then correlate the intervals between these transitions with the reference intervals for the preamble. If the transition intervals match the preamble, the receiver knows that a packet is about to begin and thus enters the next mode.
- *Packet decoding mode.* In this mode, the microcontroller again reduces the power consumption by sampling the signal only in the middle of each transmitted bit. Specifically, the microcontroller wakes up briefly to capture each sample, then sleeps until the next bit, thus saving considerable power. After the known packet length has expired, the microcontroller fully wakes up and attempts to decode the packet by performing operations such

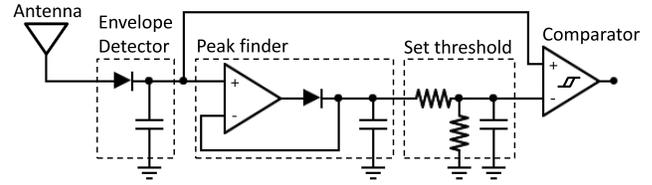


Figure 8—Receiver circuit at the tag to decode the transmissions in Fig. 7. The receiver has four main components: an envelope detector to remove the 2.4 GHz carrier frequency, a peak finder that stores the peak energy value of the Wi-Fi signals in its capacitor, a set-threshold circuit that halves the threshold values, and a comparator that outputs a one bit when the received energy is greater than the threshold value and a zero bit otherwise.

as framing and CRC checks for the Wi-Fi Backscatter messages transmitted on the downlink.

5. WI-FI BACKSCATTER IN A GENERAL WI-FI NETWORK

Typical Wi-Fi networks have multiple Wi-Fi devices that all share the same wireless medium; most of these devices are likely to be unaware of the Wi-Fi Backscatter protocol. Wi-Fi Backscatter’s downlink design addresses the problem of multiple Wi-Fi devices by using a *CTS_to_SELF* packet that prevents other Wi-Fi devices from interfering with its transmissions. The presence of multiple Wi-Fi devices, however, is problematic for the uplink design in §3.

In particular, the Wi-Fi reader uses the channel measurements from the helper’s packets to decode the information send by the Wi-Fi Backscatter tag. The assumption, however, is that for every bit sent by the Wi-Fi Backscatter tag, the Wi-Fi reader receives channel measurements from at least a few helper packets. Since Wi-Fi uses a random access MAC protocol, the number of packets per second transmitted from the Wi-Fi helper depends on the traffic in the network. Ideally, if the Wi-Fi Backscatter tag can identify the helper packets, it can ensure that there are an equal number of helper packets for each transmitted bit. This is however difficult since our Wi-Fi Backscatter tags cannot decode the Wi-Fi headers and hence cannot accurately identify the helper packets.

Wi-Fi Backscatter addresses this problem by having the Wi-Fi Backscatter device adapt its transmission rate for different network traffic loads. Specifically, the Wi-Fi reader computes the average number of packets the helper (e.g., an AP) can transmit for the current network conditions. Suppose the Wi-Fi helper can transmit, on average, N packets per second given the current network load and suppose the Wi-Fi reader requires the channel information from M packets to reliably decode each bit. Given these parameters, the rate at which the Wi-Fi Backscatter tag sends bits is given by $\frac{N}{M}$ bits per second. The Wi-Fi reader computes this bit rate and transmits this information in the query packet addressed to the Wi-Fi Backscatter tag. The latter uses this bit rate while transmitting bits on the uplink to the Wi-Fi reader. We note the following key points:

- *Dealing with bursty traffic.* While the above computation is based on the average statistics, Internet traffic in general is known for its bursty nature. Thus, it is unlikely that every bit from the Wi-Fi Backscatter tag affects the same number of helper packets. To address this problem, the Wi-Fi reader uses the timestamp information in the Wi-Fi header to bin the channel measurements to the correct bit boundaries. Further, the Wi-Fi reader provides conservative bit rate estimates to the Wi-Fi Backscatter device to minimize the probability of not receiving channel information for some of the transmitted bits.

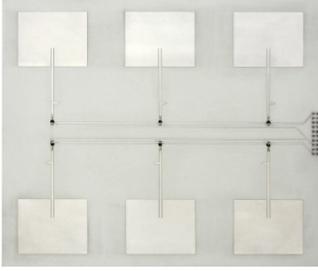


Figure 9—Wi-Fi Backscatter prototype antenna. The prototype antenna is designed to operate across the 2.4 GHz Wi-Fi channels and can both modulate the Wi-Fi channel as well as harvest RF signals. Our antenna consists of an array of six small micro-strip patch elements, each with dimensions 40.6 by 30.9 mm. We note that in principle, one can decrease the size of this antenna by using a number of other topologies that can achieve similar qualities. We use this topology due to its fabrication simplicity.

- *Leveraging traffic from all Wi-Fi devices.* In general, the Wi-Fi reader can leverage transmissions from all Wi-Fi devices in the network and combine the channel information across all of them to achieve a high data rate in a busy network. Since most of the traffic in a wireless network is downlink traffic [16], using the AP as a helper and leveraging its transmissions can eliminate the need for introducing additional traffic.
- *Using the AP’s beacon packets.* The Wi-Fi reader can use the periodic beacon packets transmitted by Wi-Fi APs to decode the bits from the tag. Such an approach, while reducing the data rates, would have minimal overhead on the Wi-Fi network throughput. In §7.5, we evaluate the feasibility of this approach.

6. PROTOTYPE IMPLEMENTATION

We build a prototype of our Wi-Fi Backscatter tag that is optimized to operate across the 2.4 GHz Wi-Fi channels. The prototype has a 2.4 GHz antenna, shown in Fig. 9, that can both modulate the Wi-Fi channel and harvest RF signals. The harvesting circuit we use is similar to those proposed in prior systems [12, 26, 23]. Our antenna is comprised of an array of six elements, each of which is a small micro-strip patch that is connected to both an RF switch and a full-wave diode rectifier that provides RF-to-DC power conversion. The ADG902 RF switch [1] from Analog Devices was selected for its broad bandwidth, low power, and good switching isolation at 2.4 GHz. Skyworks SMS7630 RF detector diodes [4] were selected for their high rectification efficiency at low RF power levels. The antenna is connected to an MSP430G2553 running custom firmware with receive and transmit logic implementations.

On the uplink, a hardware timer module of the TI MSP430 microcontroller is used to generate a bit clock and drives a simple firmware module. Each packet consists of a Wi-Fi Backscatter preamble, payload and a postamble. The reader uses the preamble and postamble to recover the bit clock from the transmitted signals. We use a 13-bit Barker code that is known for its good auto-correlation properties [6]. For the downlink, we implement the circuit design in §4.2 that allows us to identify the presence of Wi-Fi packets. We implement the energy saving mechanisms as described in §4.2. Each packet has a preamble, a payload, and a postamble. We note that the power consumption of our transmit circuit is $0.65 \mu\text{W}$, while that of the receiver circuit is $9.0 \mu\text{W}$. Our results show that the Wi-Fi power harvester can continuously run both the transmitter and receiver from a distance of one foot from the Wi-Fi reader. Additionally, in a dual-antenna system with both Wi-Fi and TV harvesting, the full system could be powered with a duty cycle

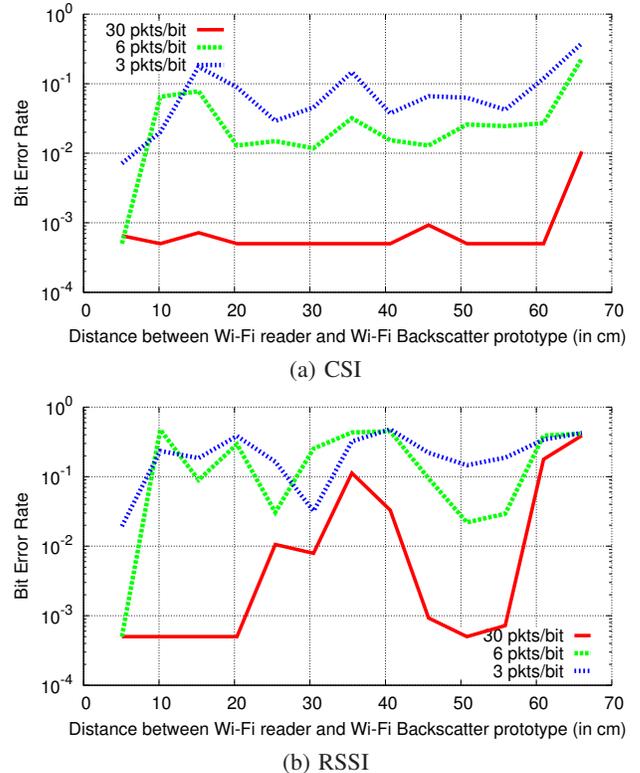


Figure 10—Effect of distance on the uplink Bit Error Rate. Figure (a) and (b) show the results when the Wi-Fi reader decodes using the CSI and RSSI information respectively. The figure also plots the BER for different number of packets per transmitted bit. The plots show that as the average number of Wi-Fi packets per bit increases, both the BER and the range improve. Further, the CSI measurements perform better than the RSSI. Finally, the receiver can decode with BERs less than 10^{-2} up to ranges of about 65 cm and 30 cm using CSI and RSSI respectively.

of around 50% at a distance of 10 km from a TV broadcast tower, independent of the distance from the Wi-Fi reader.

7. UPLINK EVALUATION

First, we evaluate Wi-Fi Backscatter’s uplink communication from our prototype device to the Wi-Fi reader. We measure the impact of various parameters including the distance between our prototype and the Wi-Fi reader, the transmission rate of the Wi-Fi helper, and its distance from the prototype device.

7.1 Uplink BER versus Distance

We compute the uplink bit error rate (BER) observed at the Wi-Fi reader as a function of the distance from the prototype device. In this section, we focus on the efficiency of our decoding algorithm from §3.2 that is designed to operate at short ranges. Later in §10, we evaluate the long range uplink communication design from §3.4.

Experiments. We use Intel Link 5300 cards as both our Wi-Fi helper and reader devices. The devices are configured to run on Wi-Fi channel 6 in the 2.4 GHz range. The results for the other 2.4 GHz Wi-Fi channels are similar and are not presented for lack of space. We inject traffic from the Wi-Fi helper and configure the Wi-Fi reader to be in monitor mode. The helper is placed three meters away from the prototype device. The reader collects the CSI information for the helper’s packets, using the Intel CSI Tool toolkit

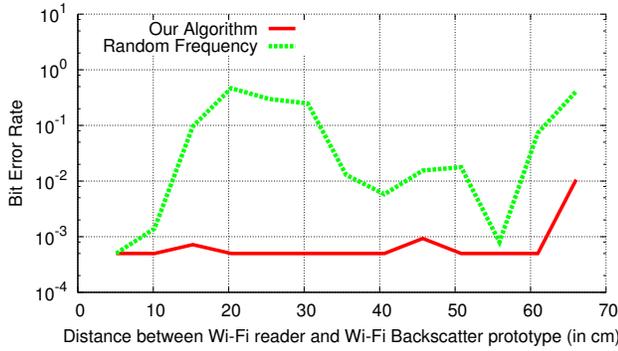


Figure 11—Effect of frequency diversity on BER. The figure shows the BER for two algorithms: one that randomly picks a Wi-Fi sub-channel and decodes, and the second is Wi-Fi Backscatter’s algorithm that leverages frequency diversity to combine information across all the sub-channels. The figure shows that leveraging frequency diversity provides substantial benefits.

from [13] on its three antennas. The Wi-Fi reader is configured to perform the algorithm in §3.2 that first identifies the good Wi-Fi sub-channels, combines them using maximum-ratio combining, and finally uses majority voting across channel measurements to decode the bits. We note that one of the antennas on our Intel device almost always reported significantly low CSI values. To avoid introducing bias, we included the CSI measurements from this antenna in our algorithm. The prototype device is set to transmit at various distances between five centimeters and 65 centimeters from the Wi-Fi reader. In each run of the experiment, the prototype device transmits a 90-bit payload message (including the Wi-Fi Backscatter preamble). We repeat the experiment 20 times at each distance value and compute the bit error rate (BER) by comparing the received bits with those transmitted across all the packet transmissions in the location. Since we transmit a total of 1800 bits, if we do not see any bit errors, we set the BER to 5×10^{-4} . The bit error rate depends on the average number of channel measurements for each bit; thus, we measure the BERs for different average number of Wi-Fi packets from the helper we use to represent each bit.

Results. Fig. 10(a) shows the BER as a function of the distance between the Wi-Fi reader and the prototype device. Fig. 10 (b) shows the corresponding results when using the RSSI information at the Wi-Fi reader, instead of CSI. The plots show the following:

- The BER increases with the distance between the prototype device and the Wi-Fi reader. This is expected because as the distance increases, the reflections from the prototype device experience higher attenuation and hence are more susceptible to noise.
- As the number of packets per bit increases, the BER significantly reduces. This is because with more packets per bit, the Wi-Fi reader receives more channel measurements and hence can achieve higher reliability using the majority voting procedure.
- The CSI information provides higher ranges and better BERs than the RSSI. This is because CSI gives us detailed channel information in each Wi-Fi OFDM sub-channel. In contrast, RSSI is a single value averaged across all the sub-channels. Thus, CSI values have more information and hence achieve lower BERs.
- Wi-Fi Backscatter can achieve distances of up to 65 cms using an average of 30 packets/bit, with the CSI information. The RSSI information, on the other hand, provides a range of about 30 cms assuming a target BER of 10^{-2} . In §10, we show how to increase this range further using our correlation mechanism.

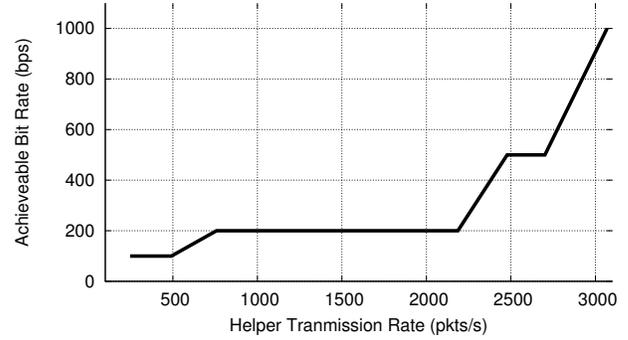


Figure 12—Uplink Bit rate versus transmission rate at the Wi-Fi helper. The average achievable bit rate is the maximum bit rate, amongst the tested rates of 100, 200, 500, and 1000 bits/s, that can be decoded at the Wi-Fi reader with a BER less than 10^{-2} . The plot shows that the bit rate is around 100 bps and 1 kbps with transmission rates of 500 and 2070 packets/second.

Effect of frequency diversity. Next we evaluate the benefits of leveraging frequency diversity across all the Wi-Fi sub-channels. Specifically, we compare two main schemes:

1. *Random-Subchannel:* We pick a random Wi-Fi sub-channel and use it to decode bits from the prototype device.
2. *Our algorithm:* We use the algorithm described in §3.2 which picks the best Wi-Fi sub-channels, combines them using maximum-ratio combining, and then decodes bits from the prototype device.

Fig. 11 shows the BER results using the two algorithms for the case where we use 30 Wi-Fi packets per bit. The plot shows that using a random Wi-Fi sub-channel performs poorly and does not operate reliably at distances greater than 15 centimeters. In contrast, our algorithm significantly reduces the BER and also operates at much larger distances. This demonstrates that leveraging frequency diversity provides substantial benefits for our uplink channel.

7.2 Data Rate Versus Helper’s Transmission Rate

The above set of experiments analyzes the achievable bit rate as a function of the average number of Wi-Fi packets required to represent each bit. The actual bit rate achieved, however, depends on the packet transmission rate at the Wi-Fi helper device. In this section, we measure the bit rate achieved by our system for different transmission rates at the Wi-Fi helper device.

Experiments. We fix the locations of the Wi-Fi reader (Intel Wi-Fi Link 5300 card) and the prototype device to be five centimeters away from each other. The Wi-Fi helper device (Intel Wi-Fi Link 5300 card) is again placed 3 meters away from the prototype device. We later present results for larger helper distance values. To change the number of packets transmitted per second at the helper device, we insert a delay between injected packets. In our network, which is running on the same Wi-Fi channel as our organization’s Wi-Fi device, this results in a transmission rate between 240 and 3070 packets per second. For each of these transmission rates, the prototype device transmits at four different bit rates (100, 200, 500, and 1000 bits/sec). We measure the achievable bit rate by measuring the maximum bit rate at which the computed average BER is less than 10^{-2} . We compute the average achievable bit rate by taking the mean of the achievable bit rates across multiple runs.

Results: Fig. 12 plots the achievable bit rate as a function of the Wi-Fi helper’s transmission rate (packets per second). The plot shows that as the number of packets transmitted per second at the helper

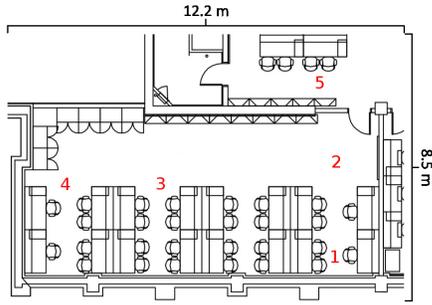


Figure 13—Testbed with the tested locations.

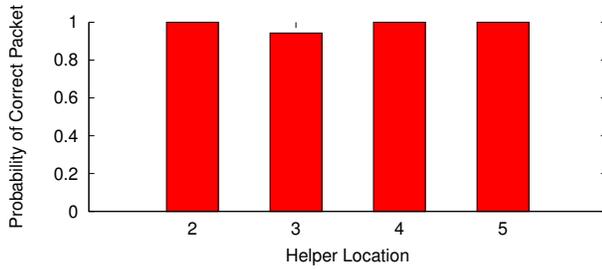


Figure 14—Probability of receiving a correct packet on the uplink for the various helper locations shown in Fig. 13. The figure shows that the Wi-Fi reader can decode packets even when the helper device is not in the same room (location 5). Further the uplink is fairly independent of the Wi-Fi helper location.

device increases, the achievable bit rate increases. The bit rate is around 100 bits/s at a helper transmission rate of 500 packets/s and is 1 kbps when the transmission rate is about 3070 packets/s. We note that these bit rates are more than sufficient for a majority of the sensing and Internet of Things applications.

7.3 Bit Rate Versus Wi-Fi Helper Location

Finally, we evaluate the effects of the Wi-Fi helper’s location on the probability of decoding correct packets from the Wi-Fi Backscatter tag at the reader. To do this, we place the prototype device and the Wi-Fi reader (Intel Wi-Fi Link 5300 card) such that they are 5 centimeters away from each other, in location 1 in Fig. 13. We then place the Wi-Fi helper device (Intel Wi-Fi Link 5300 card) in locations 2-5 that span line-of-sight and non-line-of-sight scenarios and are at distances of 3-9 meters from the tag, as shown in Fig. 13. The average CSI values span 3-50 across these locations. Note that location 5 is in a different room from our prototype device. In each of our experiments, the prototype device transmits 20 packets at a bit rate of 100 bps. We compute the average packet delivery probability, i.e., the fraction of packets received correctly at the Wi-Fi reader, at each of the above locations.

Fig. 14 plots the packet delivery probability as a function of different Wi-Fi helper device locations. The figure shows that this probability is high across all the helper locations. We also note that the Wi-Fi reader can successfully decode our packets even when the helper device is not in the same room as our prototype device. This demonstrates that the communication capabilities on the uplink are fairly independent of the Wi-Fi helper location and depend only on the distance between the Wi-Fi reader and the prototype device.

7.4 Using Only Traffic on the Network

Our experiments so far create additional traffic on the network to communicate from the prototype device to the Wi-Fi helper. We

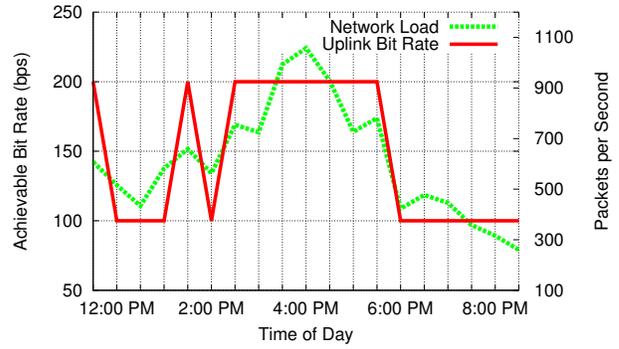


Figure 15—Achievable uplink bit rate using ambient Wi-Fi packets in an office setting as a function of time. The figure shows that the achievable bit rate is proportional to the number of packets on the network.

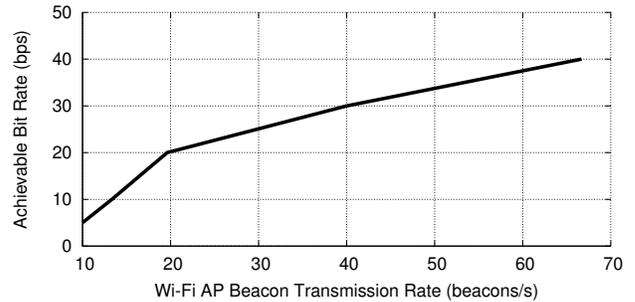


Figure 16—Achievable uplink bit rate using only the periodic beacons from the AP (Wi-Fi helper). The figure shows the feasibility of using only the beacon packets to establish the uplink.

evaluate the feasibility of uplink communication without the need for this additional traffic. Specifically, we run experiments in a lab environment in our organization by configuring the Wi-Fi helper to be in monitor mode and capturing all the packets transmitted by the organization’s AP. We place the Wi-Fi reader (an Intel Wi-Fi Link 5300 card) at a fixed distance of five centimeters away from the prototype transmitter. We run experiments once every 10 minutes and compute the achievable uplink bit rate (the maximum rate at which the bit error rate at the Wi-Fi helper is less than 10^{-2}). We also log the average number of Wi-Fi packets in the network as seen by our Wi-Fi reader device.

Fig. 15 plots the achievable bit rate as a function of time. For reference, we also plot the average number of all Wi-Fi packets as seen by the Wi-Fi helper device. The figure shows that the achievable bit rate is proportional to the number of packets on the network. This is because the better the Wi-Fi network utilization, the more the opportunities for the Wi-Fi reader to receive packets from the AP. Since the uplink bit rate depends on the transmission rate of the Wi-Fi helper device (which in our case is the AP), a better-utilized network results in higher data rates. The key point here is that we can establish the uplink communication channel without introducing additional traffic on the network.

7.5 Using only Beacon Packets

Finally, we check the feasibility of using only the periodic beacon messages from the AP. We use an Intel Wi-Fi Link 5300 card configured as the AP and an Intel Wi-Fi Link 5300 card as a Wi-Fi reader. The reader does not generate any traffic on the network and passively listens to the beacon messages periodically transmitted by the access point. No other device is associated with the access point,

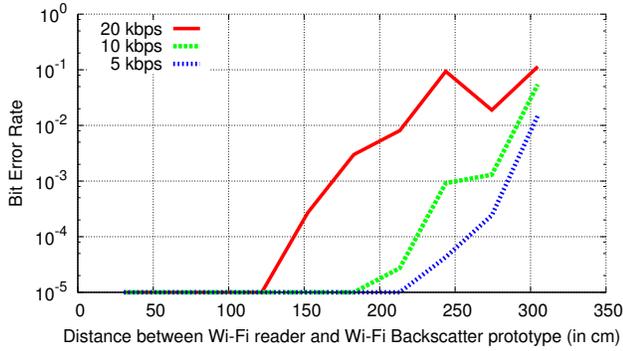


Figure 17—Downlink Bit Error Rate Versus distance between Wi-Fi Backscatter tag and Wi-Fi reader. The three bit rates correspond to Wi-Fi packets of length $50 \mu\text{s}$, $100 \mu\text{s}$, and $200 \mu\text{s}$. The figure shows that at a target BER of 10^{-2} , the prototype device can achieve bit rates of 20 kbps at distances of 2.13 m. The range can be increased to 2.90 m by decreasing the bit rate to 10 kbps.

but it operates on channel 6, which is the same frequency as our organization’s Wi-Fi network. We run experiments from 2-3 PM on a weekday. We place the Wi-Fi Backscatter tag 5 centimeters away from the reader and compute the achievable bit rate by measuring the maximum rate at which the reader can decode the tag’s transmissions at a BER less than 10^{-2} . Since Intel cards do not currently provide CSI information for beacon packets, we again use RSSI for these experiments. We repeat the experiments for different beacon frequencies at the Wi-Fi AP. Fig. 16 shows that as expected the achievable bit rate increases with the beacon frequency. The key takeaway from these results, however, is that Wi-Fi Backscatter can establish uplink communication using only the AP’s beacon packets and hence need no additional traffic to be generated on the network.

8. DOWNLINK EVALUATION

Next, we evaluate the performance of our downlink communication channel. Here, the Wi-Fi reader conveys information by encoding bits in the presence and absence of Wi-Fi packets. We evaluate the BER performance as well as the false-positive rate.

8.1 Downlink BER Versus Distance

We compute the bit error rate (BER) as a function of the distance between the Wi-Fi Backscatter tag and the Wi-Fi reader. At each distance, the Wi-Fi reader transmits a total of 200 kilobits to the Wi-Fi Backscatter tag across multiple transmissions. The transmit power at the reader is set to +16 dBm (40 mW). In each transmission, the reader encodes a ‘1’ bit as the presence of a packet and a ‘0’ bit as its absence. We run experiments with three different packet sizes of $50 \mu\text{s}$, $100 \mu\text{s}$, and $200 \mu\text{s}$ at the reader, corresponding to bit rates of 20 kbps, 10 kbps, and 5 kbps. The Wi-Fi Backscatter tag uses its thresholding circuit as described in §4.2 to decode the transmitted bits. We then compute the BER by comparing the decoded bits with the transmitted bits.

Fig. 17 plots the BER as a function of the distance. The plot shows the following:

- As expected, the BER increases with the distance between the Wi-Fi Backscatter tag and the reader. Similarly, the BER values are better at lower bit rates.
- Our prototype can efficiently identify Wi-Fi packets as small as $50 \mu\text{s}$ from nearby devices.
- At a target BER of 10^{-2} , the Wi-Fi Backscatter downlink can achieve bit rates of 20 kbps at distances of 2.13 meters between

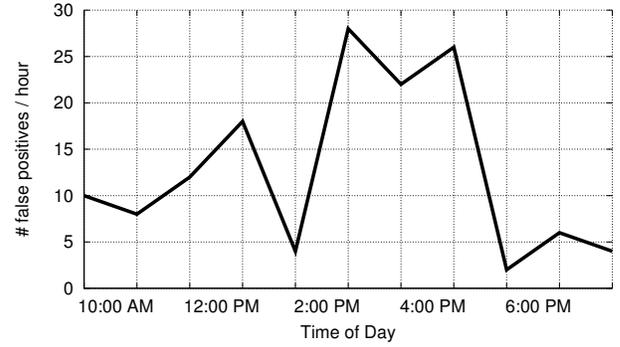


Figure 18—Downlink False-positive rate. The figure shows the number of false positive events where the prototype receiver confuses normal Wi-Fi traffic for a Wi-Fi Backscatter preamble.

the two devices. The range can be increased to 2.90 meters by decreasing the bit rate to 10 kbps.

8.2 Downlink False-Positive Rate

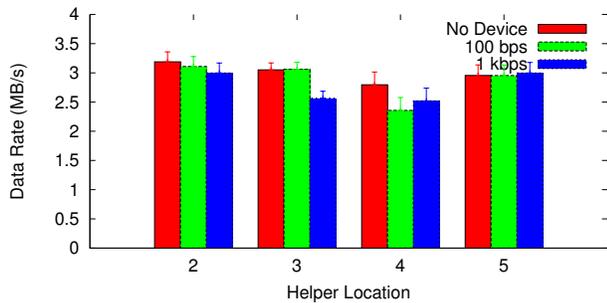
Finally, we measure the false-positive rate experienced by our downlink communication channel. We define the false-positive rate as the number of events per hour where, in the absence of a Wi-Fi Backscatter enabled Wi-Fi transmitter, our prototype device falsely detects the Wi-Fi Backscatter preamble in Fig. 7, and wakes up the microcontroller to perform decoding. To measure this, we place our receiver prototype 30 centimeters away from our network AP. To ensure that there is consistent traffic on the network, we stream music from Pandora [3] from one of the connected clients for the whole duration of the experiment. The receiver prototype is configured to log false-positive events, i.e., events when it detects the known preamble. We run our experiments during peak hours and configure our prototype to detect preambles where each bit is $50 \mu\text{s}$.

Fig. 18 shows the number of false positive events per hour at our prototype receiver, as a function of the time of the day. The figure shows that the maximum false positive rate we observe in our setup is less than 30/hour. These low numbers are expected, because it is unlikely that normal Wi-Fi traffic generates a structure that matches the Wi-Fi Backscatter preamble in Fig. 7.

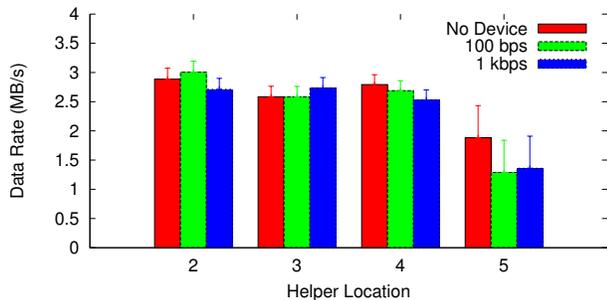
9. EFFECT OF REFLECTIONS ON WI-FI COMMUNICATION

Next, we evaluate the effects of the reflections created by our prototype device on communication between a Wi-Fi transmitter-receiver pair. Specifically, we stress-test the system when the Wi-Fi Backscatter tag is at close distances to the receiver.¹ Note that in practice, a Wi-Fi Backscatter device modulates only when queried by a Wi-Fi reader. However to stress-test the system, we set Wi-Fi Backscatter to continuously send bits at two different data rates of 1 kbps and 100 bps. Since Wi-Fi uses bit rate adaptation, we fix the Wi-Fi receiver and the prototype in location 1 in Fig. 13 and move the Wi-Fi transmitter across the remaining locations. The devices use their default bit rate adaptation algorithms. We use built in Wi-Fi of a Lenovo Thinkpad laptop as the transmitter and a Linksys WRT54GL AP as the receiver. In each run of the experiment, the Wi-Fi transmitter sends UDP packets for two minutes, and logs the throughput observed every 500 ms. We compute the average

¹Note that we do not evaluate the effect of Wi-Fi Backscatter’s downlink design on Wi-Fi traffic. Experimental evaluation of the effect of non Wi-Fi traffic on *CTS_to_SELF* and *CTS_to_SELF* on Wi-Fi traffic has been explored in prior work [10].



(a) Distance of 5 cm



(b) Distance of 30 cm

Figure 19—Effect of Wi-Fi Backscatter Uplink on Wi-Fi data rate. The figure plots the data rates for different locations for the Wi-Fi transmitter placed in the testbed in Fig. 13. The Wi-Fi Backscatter prototype is configured to continuously modulate the Wi-Fi channel and the plots show the results for three scenarios: without the prototype, with the prototype modulating at 100 bps and 1 kbps. The plots show that Wi-Fi rate adaptation can easily adapt to the variations caused by the prototype transmitter.

throughput by computing the mean across the two minute duration. In each Wi-Fi transmitter location, we measure the average throughput both in the presence and absence of our prototype.

Fig. 19 (a) and (b) shows the average data rate as a function of the Wi-Fi transmitter location, for two different distance values between the prototype device and the Wi-Fi receiver. Each figure plots the data rates in three different scenarios: in the absence of the prototype, and when the prototype transmits at 100 bps and 1 kbps. The figures show that while there is variation in the observed data rate across these scenarios (most notably in location 5 where the experiments were performed with heavy Wi-Fi utilization due to the presence of a class in the adjacent room), they are mostly within the variance. We believe this is because Wi-Fi uses rate adaptation and can easily adapt for the small variations in the channel quality, resulting from the reflections from our prototype.

10. EVALUATING CORRELATION EFFECTS ON WI-FI BACKSCATTER UPLINK RANGE

Finally, we evaluate the feasibility of extending Wi-Fi Backscatter’s range on the uplink using the correlation technique in §3.4. Specifically, we place the Wi-Fi helper (an Intel Wi-Fi Link 5300 card) 3 meters away from the Wi-Fi reader (an Intel Wi-Fi Link 5300 card). The Wi-Fi Backscatter tag encodes zero and one bits using long code sequences of length N ; the Wi-Fi reader correlates with these code sequences to decode the bits as described in §3.4. We vary the distance between the Wi-Fi reader and the prototype

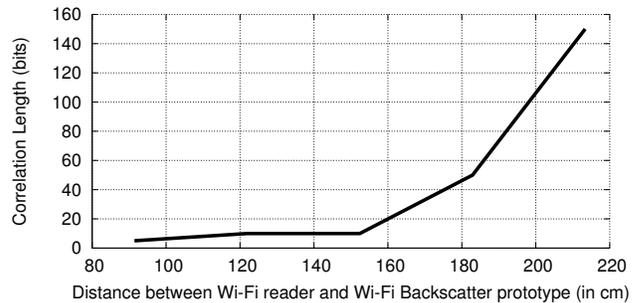


Figure 20—Increasing the uplink range using the correlation mechanism in §3.4. The figure shows that the range can be increased to more than 2 m by using long sequences to represent each transmitted bit and using correlation at the Wi-Fi reader.

and measure the correlation length at which the observed bit error rate at the Wi-Fi reader is less than 10^{-2} .

Fig. 20 plots these correlation lengths as a function of the distance between the reader and our prototype. The figure shows that as the correlation length increases, the range at which Wi-Fi Backscatter’s uplink operates also increases. Specifically, using a correlation length of 20 bits, we establish the uplink communication link at distances of about 1.6 meters between the Wi-Fi reader and the prototype. The required correlation length increases significantly with the distances between the Wi-Fi reader and the prototype. Specifically, at distances of 2.1 meters, we need a correlation length of about 150 bits. While this would reduce the effective bit rates achieved on the uplink, we emphasize that establishing a communication link between Wi-Fi Backscatter tags with existing Wi-Fi devices, albeit at a low rate, is beneficial for a large class of Internet-of-Things applications.

11. RELATED WORK

Wi-Fi Backscatter is related to work on RFID systems [8, 28, 29], which use dedicated powered infrastructure (RFID readers) to provide power and enable communication with battery-free tags. The cost of deploying and maintaining such an infrastructure has tempered the adoption of these systems. In contrast, the key value proposition of RF-powered devices is that they can harvest ambient RF signals (e.g., TV, cellular, and Wi-Fi) and thus eliminate the need for dedicated infrastructure. Since traditional radio communication consumes significantly more power than is available in RF signals [23], it has thus far been challenging to connect these devices to the Internet. Wi-Fi Backscatter addresses this problem with a novel system that bridges RF-powered devices and the Internet.

Wi-Fi Backscatter is also related to recent work on ambient backscatter communication [17] that enables two RF-powered devices to communicate by scattering ambient TV signals. While ambient backscatter can enable a network of RF-powered devices to communicate with each other, it does not provide Internet connectivity. A naive option is to deploy powered infrastructure devices that are equipped with both ambient backscatter communication and traditional power-consuming radios, but this diminishes the key benefit of RF-powered systems; an ability to operate without dedicated infrastructure. Wi-Fi Backscatter enables RF-powered devices to communicate with existing Wi-Fi infrastructure, bringing us closer to the vision of an RF-powered Internet of Things.

Wi-Fi Backscatter also differs from both RFID and ambient backscatter systems in that these systems decode backscatter information from a single continuous signal source, i.e., an RFID reader or a TV tower. Further, the decoding is performed on custom hardware that is specially designed for this purpose. In con-

trast, this paper introduces a method to modulate the Wi-Fi channel and demonstrates that we can perform decoding on off-the-shelf Wi-Fi devices. Further, we show how to detect Wi-Fi packets and communicate using such a capability.

Finally, recent work has demonstrated the ability to harvest power from ambient signals including TV [26, 19, 15, 25], and cellular transmissions [27, 23]. More recently, researchers have demonstrated the ability to harvest energy from Wi-Fi transmissions [22, 21, 11]: [14] harvests and backscatters signals using transmissions from Agilent 89600 custom 2.4 GHz transceiver hardware and [22, 11] show the feasibility of power harvesting using signals from off-the-shelf Wi-Fi access points. Wi-Fi Backscatter builds on this work but is complimentary in that it transforms existing Wi-Fi signals into a communication medium for battery-free devices. Specifically, we are the first to establish a communication link between RF-powered devices and commodity Wi-Fi devices.

12. CONCLUSION

RF-powered devices hold the promise to realize a pervasive vision of the “Internet of Things” where devices may be embedded into everyday objects and can achieve computation, sensing, and communication, all without the need to ever plug them in or maintain batteries. This paper provides the critical component in this vision of RF-powered Internet of Things: an ability to connect RF-powered devices to the Internet.

We present Wi-Fi Backscatter, a novel communication system that bridges RF-powered devices with the Internet. We show that it is possible to reuse existing Wi-Fi infrastructure to provide Internet connectivity to RF-powered devices. We show the feasibility of our approach by building a hardware prototype and demonstrating the first communication link between an RF-powered device and commodity Wi-Fi devices. We run experiments with off-the-shelf Wi-Fi devices and achieve communication rates of up to 1 kbps and ranges of up to 2.1 m. We believe that this new capability is critical for the commercial adoption of RF-powered Internet of Things.

Acknowledgements. We would like to thank Ben Ransford, Vincent Liu, Rajalakshmi Nandakumar, Donny Huang, our shepherd Kyle Jamieson, and the anonymous SIGCOMM reviewers for their helpful comments. This research is funded in part by UW Commercialization Gap Fund, Qualcomm Innovation Fellowship, Washington Research Foundation gift, NSF, and University of Washington.

13. REFERENCES

- [1] ADG902 RF switch datasheet. http://www.analog.com/static/imported-files/data_sheets/adg901_902.pdf.
- [2] Epc class 1 generation 2 uhf rfid air interface specification. http://www.gs1.org/gsm/kc/epcglobal/uhf1g2/uhf1g2_1_2_0-standard-20080511.pdf.
- [3] Pandora. www.pandora.com.
- [4] SMS7630 surface mount mixer and detector schottky diodes. http://www.skyworksinc.com/uploads/documents/Surface_Mount_Schottky_Diodes_200041W.pdf.
- [5] Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac)and physical layer (phy) specifications amendment 5: Enhancements for higher throughput. IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009), pages 1–565, Oct 2009.
- [6] R. Barker. Group synchronizing of binary digital sequences. *Communication Theory*, pages 273–287, 1953.
- [7] D. Brennan. On the maximal signal-to-noise ratio realizable from several noisy signals. *Proc. IRE*, 43(10):1530, 1955.
- [8] M. Buettner, R. Prasad, A. Sample, D. Yeager, B. Greenstein, J. R. Smith, and D. Wetherall. Rfid sensor networks with the intel wisp. In 6th ACM Conference on Embedded Network Sensor Systems, *SenSys '08*, pages 393–394, New York, NY, USA, 2008. ACM.
- [9] N. Gershenfeld, R. Krikorian, and D. Cohen. The internet of things. Scientific American.
- [10] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi. Secure in-band wireless pairing. In *Proceedings of the 20th USENIX Conference on Security, SEC'11*, pages 16–16, Berkeley, CA, USA, 2011. USENIX Association.
- [11] K. Gudan, S. Chemishkian, J. Hull, M. Reynolds, and S. Thomas. Feasibility of wireless sensors using ambient 2.4ghz rf energy. In *Sensors, 2012 IEEE*, pages 1–4, Oct 2012.
- [12] J. Hagerty, F. Helmbrecht, W. McCalpin, R. Zane, and Z. Popovic. Recycling ambient microwave energy with broad-band rectenna arrays. *Microwave Theory and Techniques, IEEE Transactions on*, 52(3):1014–1024, March 2004.
- [13] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review*, 41(1):53–53, 2011.
- [14] H. Ishizaki, H. Ikeda, Y. Yoshida, T. Maeda, T. Kuroda, and M. Mizuno. A battery-less wifi-ber modulated data transmitter with ambient radio-wave energy harvesting. In *VLSI Circuits (VLSIC), 2011 Symposium on*, pages 162–163, June 2011.
- [15] Y. Kawahara, H. Lee, and M. M. Tentzeris. Sensprout: Inkjet-printed soil moisture and leaf wetness sensor. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12*, pages 545–545, New York, NY, USA, 2012. ACM.
- [16] S. W. Kim, B.-S. Kim, and Y. Fang. Downlink and uplink resource allocation in ieee 802.11 wireless lans. *Vehicular Technology, IEEE Transactions on*, 54(1):320–327, 2005.
- [17] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient backscatter: wireless communication out of thin air. In *SIGCOMM*, 2013.
- [18] D. Masotti, A. Costanzo, and S. Adami. Design and realization of a wearable multi-frequency rf energy harvesting system. In *Antennas and Propagation (EUCAP), Proceedings of the 5th European Conference on*, pages 517–520, April 2011.
- [19] C. Mikeka, H. Arai, A. Georgiadis, and A. Collado. Dtv band micropower rf energy-harvesting circuit architecture and performance analysis. In *RFID-Technologies and Applications (RFID-TA), 2011 IEEE International Conference on*, pages 561–567, Sept 2011.
- [20] H. Ochiai and H. Imai. On the distribution of the peak-to-average power ratio in ofdm signals. *Communications, IEEE Transactions on*, 49(2):282–289, Feb 2001.
- [21] U. Olgun, C.-C. Chen, and J. Volakis. Wireless power harvesting with planar rectennas for 2.45 ghz rfids. In *Electromagnetic Theory (EMTS), 2010 URSI International Symposium on*, pages 329–331, Aug 2010.
- [22] U. Olgun, C.-C. Chen, and J. Volakis. Design of an efficient ambient wifi energy harvesting system. *Microwaves, Antennas Propagation, IET*, 6(11):1200–1206, August 2012.
- [23] A. N. Parks, A. P. Sample, Y. Zhao, and J. R. Smith. A wireless sensing platform utilizing ambient RF energy. In *IEEE Topical Meeting on Wireless Sensors and Sensor Networks (WiSNet 2013)*, January 2013.
- [24] R. Raghavendra, J. Padhye, R. Mahajan, and E. Belding. Wi-fi networks are underutilized. Technical report, Technical report, MSR, 2009.
- [25] A. Sample and J. Smith. Experimental results with two wireless power transfer systems. In *Radio and Wireless Symposium, 2009. RWS '09. IEEE*, pages 16 –18, jan. 2009.
- [26] R. Shigeta, T. Sasaki, D. M. Quan, Y. Kawahara, R. Vyas, M. Tentzeris, and T. Asami. Ambient rf energy harvesting sensor device with capacitor-leakage-aware duty cycle control. *Sensors Journal, IEEE*, 13(8):2973–2983, Aug 2013.
- [27] H. Visser, A. Reniers, and J. Theeuwes. Ambient rf energy scavenging: Gsm and wlan power density measurements. In *Microwave Conference, 2008. EuMC 2008. 38th European*, pages 721–724, Oct 2008.
- [28] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk. Efficient and reliable low-power backscatter networks. In *SIGCOMM*, 2012.
- [29] P. Zhang and D. Ganesan. Enabling bit-by-bit backscatter communication in severe energy harvesting environments. In *NSDI*, Berkeley, CA, 2014. USENIX.