Passive Wi-Fi: Bringing Low Power to Wi-Fi Transmissions

Bryce Kellogg[†], Vamsi Talla[†], Shyamnath Gollakota and Joshua R. Smith
University of Washington
[†]Co-primary Student Authors

Abstract – Wi-Fi has traditionally been considered a power-consuming communication system and has not been widely adopting in the sensor network and IoT space. We introduce *Passive Wi-Fi* that demonstrates for the first time that one can generate 802.11b transmissions using backscatter communication, while consuming 3–4 orders of magnitude lower power than existing Wi-Fi chipsets. Passive Wi-Fi transmissions can be decoded on any Wi-Fi device including routers, mobile phones and tablets. Building on this, we also present a network stack design that enables passive Wi-Fi transmitters to coexist with other devices in the ISM band, without incurring the power consumption of carrier sense and medium access control operations. We build prototype hardware and implement all four 802.11b bit rates on an FPGA platform. Our experimental evaluation shows that passive Wi-Fi transmissions can be decoded on off-the-shelf smartphones and Wi-Fi chipsets over distances of 30–100 feet in various line-of-sight and through-the-wall scenarios. Finally, we design a passive Wi-Fi IC that shows that 1 and 11 Mbps transmissions consume 14.5 and 59.2 μW respectively. This translates to 10000x lower power than existing Wi-Fi chipsets and 1000x lower power than Bluetooth LE and ZigBee.

1 Introduction

Over the past few years, researchers have explored the concept of Wi-Fi backscatter [25, 38] that creates an additional narrowband data stream to ride on top of existing Wi-Fi signals. While promising, existing designs either achieve very low data rates (100s of bps) at close by distances (2-4 feet) [25] or use custom full-duplex hardware that cannot be used with any existing Wi-Fi devices [38].

In this paper, we take a different approach — instead of backscattering existing Wi-Fi signals to send an additional data stream, we use backscatter communication to *directly generate Wi-Fi transmissions* that can be decoded on any of the billions of existing devices with a Wi-Fi chipset. To this end, we introduce Passive Wi-Fi that demonstrates for the first time that one can generate 802.11b transmissions using backscatter communication, while consuming 4–5 orders of magnitude lower power than existing Wi-Fi chipsets.

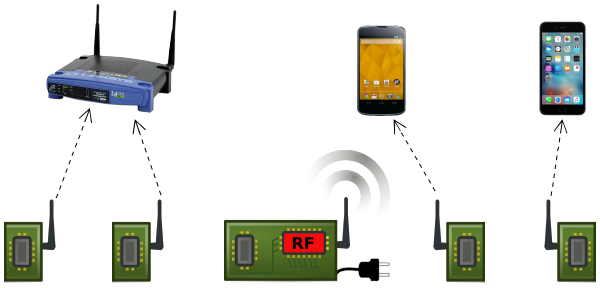


Figure 1: **Passive Wi-Fi architecture.** The passive Wi-Fi devices perform digital baseband operations like coding, while the power-consuming RF functions are delegated to a plugged-in device in the network.

We observe that while CMOS technology scaling has conventionally provided exponential benefits for the size and power consumption of digital logic systems, analog RF components, that are necessary for Wi-Fi communication, have not seen a similar power scaling. As a result, Wi-Fi transmissions on sensors and mobile devices still consume hundreds of milliwatts of power [31–33]. To get around this problem, passive Wi-Fi uses backscatter to decouple the baseband Wi-Fi digital logic from the power-consuming RF components, as shown in Fig. 1.

In our architecture, the passive Wi-Fi devices perform digital baseband operations like coding and modulation, while the power-consuming RF components such as frequency synthesizers and power amplifiers are delegated to a single plugged-in device in the network. This device provides the RF functions for all the passive Wi-Fi devices in the vicinity by transmitting a single-frequency tone. The passive Wi-Fi devices create 802.11b transmissions by reflecting or absorbing this tone using a digital switch running at baseband. Since the passive Wi-Fi

devices have no analog components, they consumes less silicon area and would be smaller and cheaper than existing Wi-Fi chipsets. More importantly, their power consumption would be orders of magnitude lower since they only perform digital baseband operations. To realize this, however, we need to address three main challenges.

(a) How can Wi-Fi receivers decode in the presence of interference from the plugged-in device? The Wi-Fi receiver receives the backscattered signal in the presence of a strong interference from the tone transmitted by the plugged-in device. Traditional backscatter systems [34, 38] use a full-duplex radio to cancel this strong interfering signal, which is not possible on existing Wi-Fi devices. Our key observation is that Wi-Fi receivers are required to work even in the presence of interference in the adjacent band that is 35 dB stronger [12]. Further, as Wi-Fi and Bluetooth radios are being integrated onto a single chipset [6], Wi-Fi hardware is being designed to work in the presence of out-of-band Bluetooth interference. Thus, we set the plugged-in device to transmit its tone at a frequency that lies outside the desired Wi-Fi channel; this ensures that existing Wi-Fi chipsets can suppress the resulting out-of-band interference.

(b) How can we create 802.11b transmissions using backscatter? At a high level, we first shift the out-of-band tone from the plugged-in device to lie at the center of the desired Wi-Fi channel. We then use this shifted tone to create 802.11b transmissions. Intuitively, multiplying two sinusoidal signals can create a frequency shift.[1] Thus, by backscattering at a frequency Δf, we can shift the tone. To synthesize Wi-Fi transmissions, we leverage that 802.11b uses DSSS and CCK encoding on top of DBPSK and DQPSK modulation. The encoding operation is digital in nature and hence is achieved using digital logic. To create the phase changes required for DBPSK and DQPSK, we approximate a digital square wave as a sinusoid and modulate its phase by changing the timing of the square wave (see §2.3). Thus, passive Wi-Fi devices can fully operate in the digital domain at baseband and yet synthesize 802.11b transmissions.

(c) How do passive Wi-Fi devices share the Wi-Fi network? Traditional Wi-Fi shares the network using carrier sense. However, this requires a Wi-Fi receiver that is ON before every transmission. Since Wi-Fi receivers require power-consuming RF components such as ADCs and frequency synthesizers, this would eliminate the power savings from our design. Instead, we delegate the power-consuming task of carrier sense to the plugged-in device. At a high level, the plugged-in device performs carrier sense and signals the passive Wi-Fi device to transmit. §3 describes how such a signaling mechanism can also be used to arbitrate the channel between multiple passive Wi-Fi devices and address other link-layer issues including ACKs and retransmissions.

To show the feasibility of our design, we build prototype backscatter hardware and implement all four 802.11b bit rates on an FPGA platform. Our experimental evaluation shows that passive Wi-Fi transmissions can be decoded on off-the-shelf smartphones and Wi-Fi chipsets over distances of 30–100 feet in various line-of-sight and through-the-wall scenarios. We also design a passive Wi-Fi IC that performs 1 Mbps and 11 Mbps 802.11b transmissions and estimate the power consumption using Cadence and Synopsis toolkits [5, 19]. Our results show the 1 and 11 Mbps passive Wi-Fi transmissions consume 14.5 and 59.2 μW respectively.

Contributions. We make the following contributions:

- We demonstrate for the first time that one can generate 802.11b transmissions using backscatter communication. We present backscatter techniques that synthesize 22 MHz DSSS and CCK spread spectrum transmissions that can be decoded on existing Wi-Fi devices.
- We design a network stack for the passive Wi-Fi transmitters to coexist with other devices in the ISM band. Further, we present a detailed analytical model to understand the operational range of passive Wi-Fi transmissions in different deployment scenarios.
- We build a hardware prototype on an FPGA platform and evaluate it in various scenarios. We also design a passive Wi-Fi IC and present its power numbers.

2 Passive Wi-Fi Design

Our design has two main actors: a plugged-in device and passive Wi-Fi devices. The former contains power consuming RF components including frequency synthesizer and power amplifier and emits a single tone RF carrier. It also performs carrier sense on behalf of the passive Wi-Fi device and helps coordinate medium access control across multiple passive Wi-Fi devices. The passive Wi-Fi device backscatters the tone emitted by the plugged-in device to synthesize 802.11b transmissions that can be decoded on any device that has a Wi-Fi chipset.

In the rest of this section, we first provide a quick primer for 802.11b physical layer and backscatter communication. We then explain how the passive Wi-Fi devices generate 802.11b packets using backscatter communication. We then theoretically analyze the range of our transmissions in various deployments scenarios.

2.1 Primer for 802.11b Transmissions

802.11b is a set of Wi-Fi physical layer specifications that use spread spectrum modulation. 802.11b uses DBPSK/DQPSK at the physical layer and achieves four

[1] $2sinftsin\Delta ft = cos(f-\Delta f)t - cos(f+\Delta f)t$.

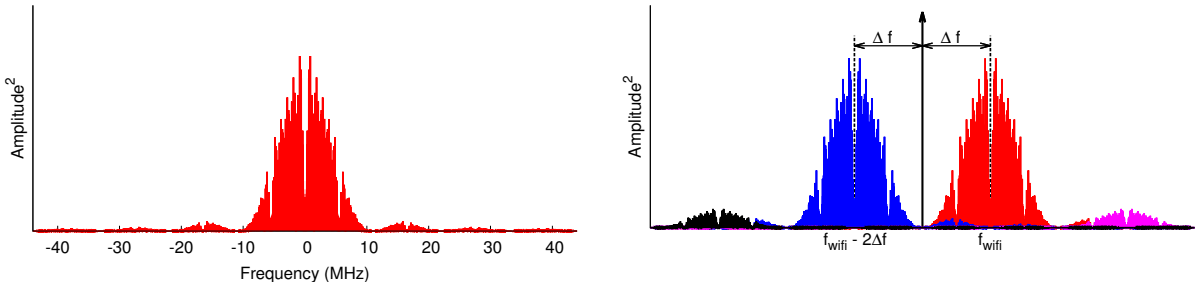


Figure 2: **Generation of Wi-Fi packets using backscatter.** The plot on the left shows the 22 MHz main lobe and the side lobes of the baseband 802.11b packet in the frequency domain. The plot on the right illustrates the backscatter operation at the passive Wi-Fi device. The two main lobes are shifted by Δf with respect to the constant tone emitted by the plugged-in device to generate the Wi-Fi packet (in red) at f_{wifi} and a mirror image (in blue) at $f_{wifi} - 2\Delta f$.

bit rates using different spreading codes. The lower two bit rates of 1 and 2 Mbps use direct-sequence spread spectrum (DSSS) while 5.5 and 11 Mbps use complementary code keying (CCK). DSSS uses a single code to spread the information over 22 MHz, while CCK uses a set of multiple code words to both encode bits and also achieve a 22 MHz spread spectrum signal. We outline how each of the 802.11b bit rates are encoded.

1 and 2 Mbps DSSS transmissions. To generate this, 802.11b first creates coded bits from the incoming data using a 11-bit barker code [39]. Specifically, 802.11b uses a single barker sequence, 10110111000, that is generated at a baseband frequency of 11 MHz to spread the spectrum over 22 MHz. To create the coded bits, 802.11b XORs each of the data bits with the barker sequence. Thus, the coded bits for a '1' data bit are 10110111000 and that for the '0' data bit are 01001000111. Each of these coded bits is encoded using DBPSK and DQPSK modulation to achieve 1 and 2 Mbps transmissions respectively. At a high level, this is achieved by setting the phase of the carrier, $sin\theta$. DBPSK modulation encodes a 0 and 1 bit by setting θ to either 0 or π, while DQPSK encodes pairs of bits by modulating the phase between 0, $\pi/2$, π and $3\pi/2$.

5.5 and 11 Mbps CCK transmissions. Instead of using a single barker code, CCK uses a set of 8-bit code words. At a high level, to generate 5.5 Mbps transmissions, the incoming data bit stream is divided into blocks of 4 bits. The first two bits are used to pick the DQPSK phase and the last two bits are used to pick a spreading code amongst four 8-bit code words. To generate 11 Mbps 802.11b transmissions, the incoming data bits are instead divided into 8 bit blocks where the first two bits determine the DQPSK phase shift and the last 6 bits are used to pick a spreading code amongst 64 8-bit code words.

To summarize, 802.11b requires generating the coded bits using either DSSS or CCK and then modulating these bits with DBPSK or DQPSK. The first operation is typically implemented in digital baseband logic while the second require changing the phase of I and Q components. Finally, we note also that since the RF energy is spread across a wide band, spread spectrum transmissions are resilient to narrowband interference both within and outside the Wi-Fi channel [39].

2.2 Backscatter Communication Primer

Unlike traditional active radio communication that requires generating RF signals, devices using backscatter communication modulate the radar cross-section of their antenna to change the reflected signal. To understand how backscatter works, consider a device that can switch the impedance of its antenna between two states. The effect of changing the antenna impedance is that the radar cross-section, i.e., the signal reflected by the antenna, also changes between the two different states. Now, given an incident signal with power $P_{incident}$, the power in the backscattered signal can be written as,

$$P_{backscatter} = P_{incident} \frac{|\Gamma_1^* - \Gamma_2^*|^2}{4} \quad (1)$$

Here Γ_1^* and Γ_2^* are the complex conjugates of the reflection coefficients corresponding to the two impedance states. Thus to maximize the power in the backscattered signal we need to maximize the difference in the power of the two impedance states which is given by $|\Delta\Gamma|^2 = \frac{|\Gamma_1^* - \Gamma_2^*|^2}{4}$. Ideally, to ensure that the power in the backscattered signal is equal to that of the incident signal, we set $|\Delta\Gamma|^2$ to 4 which can be achieved by modulating the reflection coefficients between $+1$ and -1. In practice, however, backscatter hardware deviates from this ideal behavior and incurs losses; our hardware implementation has a loss of around 1.1 dB.

2.3 802.11b using passive Wi-Fi

Generating a Wi-Fi packet using backscatter is challenging for two main reasons. First, the backscattered signal

is much weaker than the tone transmitted by the plugged-in device. A Wi-Fi receiver would suffer significant in-band interference from this tone preventing it from decoding. Second, the passive Wi-Fi device has a single digital switch that toggles between two impedance states, resulting in a binary signal. It is unclear how one may generate Wi-Fi transmissions using such a binary system.

We outline how to address these challenges. We first describe the signal transmissions from the plugged-in device and then the operations at the passive Wi-Fi device that allow us to synthesize 802.11b transmissions.

Transmissions at the plugged-in device. It transmits a tone outside the desired Wi-Fi channel. Our key intuition is that Wi-Fi receivers are designed to function in the presence of out-of-band interference: 802.11b receivers are required to ensure that the sensitivity is reduced by no more than 6 dB in the presence of interference in the adjacent band that is 35 dB greater than the in-band signal [12]. Further, as Wi-Fi and Bluetooth radios are being integrated onto the same chipset [6], Wi-Fi frontends are being designed to function in the presence of out-of-band interference from Bluetooth devices. Since the tone from the plugged-in device is narrower in bandwidth than Bluetooth, this would further help suppress the tone if it is outside the desired Wi-Fi channel.

We note however that excessive out-of-band interference, which occurs when the Wi-Fi receiver is right next to the plugged-in device, can saturate and/or compress the RF front end resulting in significant degradation of Wi-Fi performance. This is called the input 1 dB compression point which is around 0 dBm for commercial Wi-Fi devices [13]. Passive Wi-Fi inherently avoids this issue by ensuring that the Wi-Fi receiver (e.g., smartphone or router) is not next to the plugged-in device.

Backscatter operations at passive Wi-Fi devices. At a high level, the passive Wi-Fi operations can be described as first shifting the out-of-band tone transmitted from the plugged-in device to lie at the center of the desired Wi-Fi channel. We then use this shifted tone to create 802.11b transmissions. To do this, we leverage three key facts: (1) From basic trigonometry, $2sinftsin\Delta ft = cos(f-\Delta f)t - cos(f+\Delta f)t$. Thus, multiplying two sinusoidal signals can create a frequency shift. (2) Modulating the radar cross section of an antenna effectively multiplies the incoming signal by the modulated signal. Thus, modulating the antenna at a frequency Δf would create a frequency shift in the incoming signal. (3) All bit rates in 802.11b are differentially phase modulated using DBPSK or DQPSK.

Step 1. Shifting the tone from the plugged-in device using backscatter. Say the plugged-in device sends the tone $sin2\pi(f_{wifi}-\Delta f)t$ outside the Wi-Fi channel. Passive Wi-Fi devices use a square wave at a frequency of Δf to shift the tone to the center of the Wi-Fi channel. From Fourier analysis, a square wave can be written as,

$$Square(\Delta ft) = \frac{4}{\pi}\sum_{n=1,3,5,..}^{\infty}\frac{1}{n}sin(2\pi n\Delta ft)$$

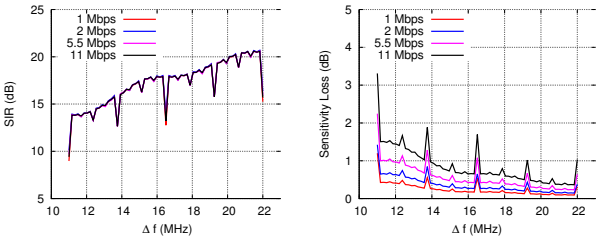
Here the first harmonic is a sinusoidal signal at the desired frequency Δf. Note that the power in each of these harmonic scales as $\frac{1}{n^2}$. So the third and the fifth harmonic are around 9.5 dB and 14 dB lower than the first harmonic. Thus, we can approximate a square wave as just the sinusoidal signal, $\frac{4}{\pi}sin(2\pi\Delta ft)$. Since modulating the radar cross section of an antenna effectively multiplies the incoming signal by the modulated signal, the backscatter signal can be approximated as $sin2\pi(f_{wifi}-\Delta f)tsin2\pi\Delta ft$. So we have used backscatter to effectively creates two tones, one centered at f_{wifi} and the other at $f_{wifi}-2\Delta f$; the first tone is at the center of the desired Wi-Fi channel.

Step 2. Synthesizing 802.11b transmissions using backscatter. Now that we have a tone centered at the Wi-Fi channel, the next step is to create 802.11b transmissions using backscatter. 802.11b uses DSSS and CCK encoding which are both digital operations and hence can be performed using digital logic at the passive Wi-Fi device. So the question that remains is: how do we generate DBPSK and DQPSK using just a square wave created at a frequency Δf by the backscatter switch?

Passive Wi-Fi does this by noting that DBPSK and DQPSK use a sine wave with four distinct phases: $0, \pi/2, \pi, 3\pi/2$. Since the square wave generated by our digital switch can be approximated as a sine wave, we can generate the required four phases by changing the timing of our square wave. Specifically, shifting the square wave by half of a symbol time, effectively creates a phase change of π. Phase changes of $\pi/2$ and $3\pi/2$ can be achieved by shifting the square wave by one-fourth and three-fourth of a symbol time. Thus, passive Wi-Fi devices can fully operate in the digital domain while run at a baseband frequency of a few tens of MHz and synthesize 802.11b transmissions using backscatter.

We note the following properties of our design.

- In addition to creating a 802.11b transmission centered at f_{wifi}, as shown in Fig. 2, our backscatter mechanism also creates a mirror copy centered at $f_{wifi}-2\Delta f$ on the other side of the tone. Thus, we use twice the bandwidth of a traditional 802.11b transmission. This is the tradeoff we make to achieve orders of magnitude lower power consumption. We note that such a tradeoff is common in 802.11n systems which use channel bonding of adjacent Wi-Fi channels to double the throughput.
- 802.11b transmissions have side lobes (Fig. 2); the side lobes of the mirror copy creates interference for the desired Wi-Fi signal. We plot the signal to interference



(a) Signal to Interference ratio *(b) Sensitivity Loss*

Figure 3: **SIR and loss in receiver sensitivity.** The plot shows the effect of different Δf's on the quality and the sensitivity of the synthesized Wi-Fi packets.

ratio for different frequency shifts, Δf, at the passive Wi-Fi device. Fig. 3(a) plots the results for all four 802.11b bit rates and shows that the interference from the side lobes of the mirror copy reduces as Δf increases. This is because, as Δf increases, the mirror copies are further separated in frequency, resulting in lower interference.

• An effect of this interference, however, is that it adds additional noise to the Wi-Fi signal, reducing the noise sensitivity at which each of the 802.11b bit rates can be decoded. Fig. 3(b) shows the loss in sensitivity for the four 802.11b bit rates, as a function of the frequency offset, Δf. The plots show that the sensitivity loss is slightly larger for higher 802.11b bit rates. This is because higher bit rates require a cleaner signal to successfully be decoded. Our system sets Δf to 12.375 MHz, where the sensitivity loss is less than 2 dB across all 802.11b bit rates. This also ensures that the passive Wi-Fi transmissions only occupy two adjacent Wi-Fi channels. Note that Wi-Fi applies filters to remove the interfering side lobes. Our implementation however does not do this.

2.4 Analyzing Passive Wi-Fi's Range

In passive Wi-Fi, the communication range depends on two parameters: the distance between the plugged-in device and the passive Wi-Fi transmitter and the distance between the passive Wi-Fi transmitter and the Wi-Fi receiver. Specifically, the signal strength at the receiver, P_r, can be modeled using Friis path loss [34] as follows,

$$P_r = \left(\frac{P_t G_t}{4\pi d_1^2}\right)\left(\frac{\lambda^2 G_{passive}^2}{4\pi}\frac{|\Delta\Gamma|^2}{4}\alpha_{wifi}\right)\left(\frac{1}{4\pi d_2^2}\frac{\lambda^2 G_r}{4\pi}\right)$$

This equation has three key parts: the term in first parenthesis models signal propagation from the plugged-in device, with an output power P_t and an antenna gain G_t, to a passive Wi-Fi transmitter at a distance d_1 away. The third term, similarly, models the signal propagation from the passive Wi-Fi transmitter to a Wi-Fi receiver with an antenna gain G_r and at a distance d_2 away. Here, λ is the wavelength of the RF signal been transmitted. Finally, the middle parenthesis models the fraction of incident signal from the plugged-in device that is backscattered by a passive Wi-Fi transmitter with an antenna gain $G_{passive}$. $|\Delta\Gamma|^2$ is the backscatter coefficient which is a measure of the efficiency with which passive Wi-Fi can generate backscatter signals. As described in §2.2, this is 1.1 dB in our hardware. Finally, α_{wifi} models the loss in energy due to synthesis of Wi-Fi signals using backscatter. This is around 4.4 dB and includes half the power lost in the mirror copy generated by backscatter and the losses due to the side lobes as described in §2.3.

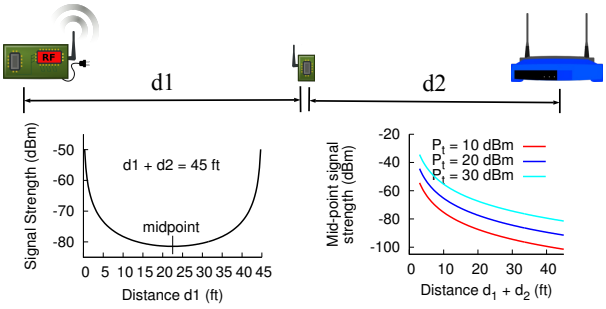


Figure 4: **Passive Wi-Fi's analytical received signal strength.** The passive Wi-Fi device moves along the line connecting the Wi-Fi router and plugged-in device.

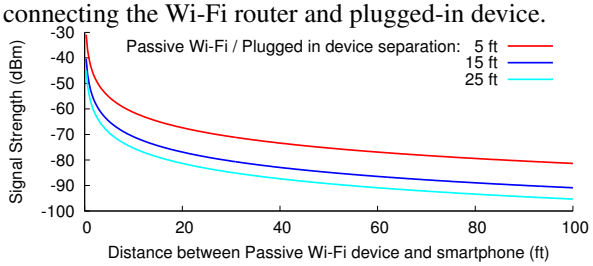
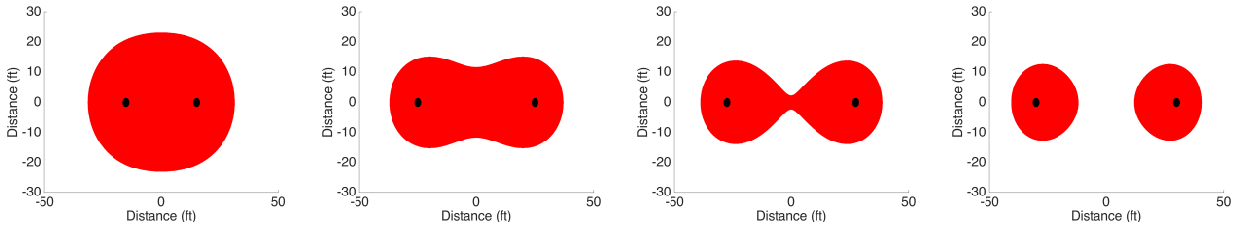


Figure 5: **Signal strength versus distance between passive Wi-Fi transmitter and Wi-Fi receiver.**

To gain a better intuition, consider the scenario in Fig. 4 where we place the plugged-in device and the Wi-Fi receiver separated by 45 feet. We move the passive Wi-Fi transmitter between these devices, along the line connecting them. We set P_t, G_t, G_r and $G_{passive}$ to 30 dBm, 6 dBi, 0 dBi, and 2 dBi respectively. Fig. 4 shows the received signal strength, P_r, as we move the passive Wi-Fi transmitter between the plugged-in device and the Wi-Fi receiver. The plots show two key points.

(1) The received signal increases as the passive Wi-Fi transmitter gets close to either the Wi-Fi receiver or the plugged-in device. This is because, maximizing the signal strength requires minimizing the product $d_1 d_2$, which is achieved either by reducing the distance d_1 or d_2.

(2) The mid-point between the plugged-in device and Wi-Fi receiver has the lowest strength. Fig. 4 shows this mid-point signal strength, as we change the distance between the plugged-in device and Wi-Fi receiver. The plot shows that this decreases with distance between the plugged-in device and the Wi-Fi receiver. As expected, it increases with plugged-in device's transmit power (P_t).



(a) 30 ft Separation (b) 50 ft Separation (c) 55 ft Separation (d) 60 ft Separation

Figure 6: **Theoretical coverage maps for different distances between the plugged-in device and the Wi-Fi router.** The black dots denote the positions for these devices. The red region represents points in the 2D space where a passive Wi-Fi transmitter can be located, while ensuring that the signal from it to the Wi-Fi router is at least -85 dBm.

2.4.1 Understanding Deployment Scenarios

1. I want to deploy passive Wi-Fi devices in my home. Where do I place the plugged-in device so as to maximize their range? Fig. 5 shows the theoretical signal strength at the Wi-Fi receiver as a function of its distance from the passive Wi-Fi transmitter. We show the results for different distances between the passive Wi-Fi transmitter and the plugged-in device. We set G_t, G_r, $G_{passive}$, P_t to 6 dBi, 0 dBi, 2 dBi, and 30 dBm respectively. The plot shows that, in general, as the distance between the passive Wi-Fi transmitter and Wi-Fi receiver increases, the received signal strength reduces. More importantly, as the distance between the passive Wi-Fi transmitter and plugged-in device decreases, the coverage range increases. This is because, from our analysis, the signal strength can be increased either by reducing the distance between the passive Wi-Fi transmitter and the plugged-in device or that between the passive Wi-Fi transmitter and the Wi-Fi receiver. Since our goal is to maximize range, we should reduce the distance between the passive Wi-Fi transmitter and the plugged-in device. In the presence of multiple passive Wi-Fi devices, this would translate to minimizing the worst-case distance between the plugged-in device and all passive Wi-Fi transmitters.

2. Where do I place my Wi-Fi router and the plugged-in device, so that I can have passive Wi-Fi devices work from anywhere in my home? Fig. 6 shows the 2D coverage maps for different distances between the plugged-in device and the Wi-Fi router. The red region represent points in the 2D space where a passive Wi-Fi transmitter can be located, while ensuring that the signal from it to the Wi-Fi router is at least -85 dBm. These maps show that the coverage area is a union of two circles centered each at the Wi-Fi router and the plugged-in device. So, as a general rule of thumb, it is better to deploy the plugged-in device and the Wi-Fi router at either ends of the coverage area. Note however that at very large distances between the plugged-in device and Wi-Fi router (Figs. 6 (c) and (d)), we end up getting two islands of coverage. Such large distance deployments are suitable only when the passive Wi-Fi transmitters are going to be close to either the plugged-in device or the Wi-Fi router.

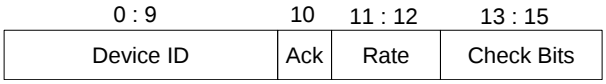


Figure 7: **Structure of the signaling packet.**

3 Passive Wi-Fi Network Stack Design

We first describe how passive Wi-Fi devices share the ISM band. We then address the issue of ACKs and retransmissions and finally, present our protocol to associate passive Wi-Fi devices with the network.

3.1 Sharing the ISM band

Wi-Fi uses carrier sense to share the ISM band. This however requires a Wi-Fi receiver that is ON before every transmission. Since Wi-Fi receivers require power-consuming RF components like LNA, frequency synthesizers, mixers and ADCs, this would eliminate the power savings from our design. Instead, we delegate the task of carrier sense to the plugged-in device, which also arbitrates access between multiple passive Wi-Fi devices.

We illustrate this with an example. Say a passive Wi-Fi transmitter wants to sent a packet on channel 6 and the plugged-in device transmits its tone between Wi-Fi channels 1 and 6. Before any of the above transmissions happen, the plugged-in device first uses carrier sense to ensure that there are no ongoing transmissions on any the frequencies including and in between channel 1 and 6.

Once the channels are found free, the plugged-in device sends a packet signaling a specific passive Wi-Fi device to transmit. This signal is sent and decoded using the ultra-low power receiver described in §3.1.1. The packet starts with an ID unique to each passive Wi-Fi device (see Fig. 7). When the passive Wi-Fi device detects its ID, it transmits within a SIFS duration at the end of the signaling packet. The signaling packet is sent at the center of channel 1 and 6 as well as in between them. This prevents other devices in the ISM band from capturing the channel before the passive Wi-Fi device gets to transmit. The packet has 16 bits and adds a fixed overhead of 100 μs for every passive Wi-Fi transmission.

The above description assumes that the plugged-in device knows when to send the signaling packet to each of

the passive Wi-Fi devices in the network. To see how this can be achieved let us focus on our target IoT applications. A device sending out beacons is configured to send them at a fixed rate. Temperature sensors, microphones and Wi-Fi cameras (e.g., Dropcam [8]) have a fixed rate at which they generate data. Similarly, motion sensors have an upper bound on the delay they can tolerate. The passive Wi-Fi devices convey this information to the plugged-in device during association (and can update it later using the protocol in §3.3). This information is used by the plugged-in device to signal each passive Wi-Fi device in accordance to its desired update rate.

3.1.1 Ultra-low power receiver design

We encode bits using ON-OFF keying. We use a passive energy detector with analog components and a comparator to distinguish between the presence and absence of energy. Our design is the same as that used in our prior work [25, 26] and we skip it for brevity. We implement the receiver using off-the-shelf components and it consumes 18 μW, while achieving a bit rate of 160 kbps.

3.2 ACKs and Rate Adaptation

ACKs and retransmissions. The plugged-in device listens to the ACKs and conveys this information back to the passive Wi-Fi sensor. Specifically, if the ACK is successfully decoded at the plugged-in device, it sets the ACK bit in the signaling packet shown in Fig. 7 to 1 and sends it to the passive Wi-Fi sensor, by piggybacking it during the next period when the sensor is scheduled to transmit. If the ACK is not received at the plugged-in device, it immediately performs carrier sense and sends a signaling packet with the ACK bit set to 0. When the passive Wi-Fi sensor receives this, it retransmits its sensor value. In our implementation, the plugged-in device detects an ACK by detecting energy for a ACK duration at the end of the passive Wi-Fi transmission.

Rate adaptation. Wi-Fi bit rate adaptation algorithms typically use packet loss as a proxy to adapt the transmitter bit rate. In our design, we delegate this function to the plugged-in device. Specifically, the plugged-in device estimates the packet loss rate for each of its associated passive Wi-Fi devices by computing the fraction of successfully acknowledged packets. It then estimates the best 802.11b bit rate and encodes this information in the bit rate field of the signaling packet. Since the plugged-in device knows the bit rate as well as the packet length (from association as described in §3.3), it knows how long the transmissions from each of its passive Wi-Fi devices would occupy on the wireless medium. Thus, it stops transmitting its tone at the end of the passive Wi-Fi transmission and listens for the corresponding ACKs.

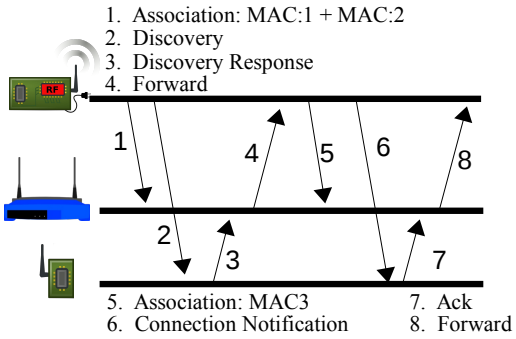


Figure 8: **Passive Wi-Fi association procedure.**

3.3 Network Association

Finally, we describe how the passive Wi-Fi transmitters associate with the plugged-in device as well as with the Wi-Fi router in the network. The key challenge is that since the plugged-in device does not have a full-duplex radio (the lack of which is desirable to make it practical and keep it low cost), there is no direct communication channel from the passive Wi-Fi device to the plugged-in device. Instead, as shown in Fig. 8, the plugged-in device associates with the Wi-Fi router with two MAC address (MAC:1 and MAC:2). The plugged-in device then broadcasts a discovery packet using ON-OFF keying modulation that contains these two MAC addresses and starts with a broadcast ID. The new passive Wi-Fi device then transmits a Wi-Fi packet with the source and destination addresses set to MAC:2 and MAC:1; this packet gets routed through the Wi-Fi router to the plugged-in device. The packet payload includes the sensor update rate, packet length, supported bit rates and its MAC address, MAC:3. The plugged-in device spoofs MAC:3 and associates it with the Wi-Fi router. It then picks a unique ID and sends it to the passive Wi-Fi device along with other Wi-Fi network credentials. Finally, the passive Wi-Fi device responds with a Wi-Fi packet with the source and destination addresses set to MAC:3 and MAC:1; this packet gets routed through the Wi-Fi router and confirms association at the plugged-in device.

After association, the passive Wi-Fi transmitter can send Wi-Fi packets to the plugged-in device through the router, and change its parameters including update rate and packet length. Note that the credentials for the spoofed MAC addresses could be sent securely using a manufacturer set secret key shared between the passive Wi-Fi devices and the plugged-in devices.

4 Hardware Implementation

We first describe our implementation of passive Wi-Fi using off-the shelf components on an FPGA platform. We use this to characterize passive Wi-Fi in various deployment scenarios. We then present our IC design

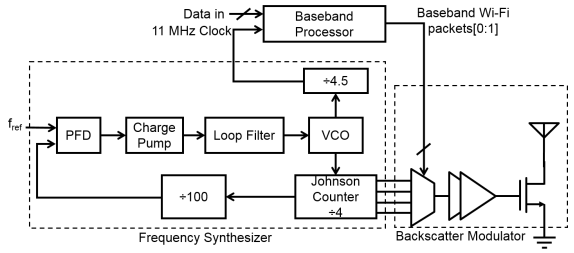


Figure 9: **Passive Wi-Fi's IC architecture.** The frequency synthesizer generates baseband clock.

which we use to quantify our power consumption.

Off-the-shelf implementation. We implement a passive Wi-Fi prototype using off-the-shelf components for backscatter and an FPGA for digital processing. The backscatter modulator consisted of HMC190BMS8 SPDT RF switch network on a 2-layer Rodgers 4350 substrate [10]. The switch was designed to modulate between open and closed impedance states and had a 1.1 dB loss. All the required baseband processing including data scrambling, header generation, DSSS/CCK encoding, CRC computation and DBPSK/DQPSK modulation were written in Verilog. The Verilog code was synthesized and programmed on a DE1 Cyclone II FPGA development board by Altera [2]. We implement four shifts of 12.375, 16.5, 22 and 44 MHz. The digital output of the FPGA was connected to the backscatter switch to generate the Wi-Fi packets. A 2 dBi omnidirectional antenna was used on the passive Wi-Fi device. The plugged-in device was set to transmit at an EIRP of 30 dBm.

Integrated circuit implementation. CMOS technology scaling has enabled the exponential scaling in power and area for integrated circuits. Wi-Fi chipsets have tried to leverage scaling but with limited success due to the need for power hungry analog components that do not scale in power and size with CMOS technology. However, baseband Wi-Fi operations are implemented in the digital domain and tend to scale very well with CMOS. For context, Atheros's AR6003 [4] and AR9462 [17] chipsets that were released in 2009 and 2012 use 65 nm CMOS and 55 nm CMOS node implementations respectively. For passive Wi-Fi device's integrated circuit implementation, we chose the 65 nm LP CMOS node by TSMC, which gives us power savings of baseband processing and ensures a fair comparison with current industry standards. The IC architecture of the passive Wi-Fi device is shown in Fig. 9 and has three main components:

Baseband frequency synthesizer. It generates the 11 MHz clock required for baseband processing as well as four phases at 12.375 MHz offsets required for DBPSK and DQPSK. We phase synchronize the 11 MHz and 12.375 MHz clocks to avoid glitches during phase modulation. We used an integer N charge pump and ring oscillator-based PLL to generate 49.5 MHz clock from a 12.375 kHz reference. The 49.5 MHz clock is fed to a quadrature Johnson counter to generate the four phases with the required timing offsets (corresponding to 0, $\frac{\pi}{2}$, π and $\frac{3\pi}{2}$ phases). The same 49.5 MHz carrier is divided by 4.5 to generate the 11 MHz baseband clock.

Table 1: **Passive Wi-Fi's IC Power Consumption**

	1 Mbps	**11 Mbps**
Baseband Frequency Synthesizer	5.6 μW	5.6 μW
Baseband Processor	5.0 μW	48 μW
Backscatter Modulator	3.9 μW	5.6 μW
Total Power	14.5 μW	59.2 μW

Baseband processor. It takes the payload bits as input and generates baseband 802.11b Wi-Fi packet. We used the Verilog code that was verified on the FPGA and use the Design Compiler by Synopsis to generate the transistor level implementation of the baseband processor [19].

Backscatter modulator. It mixes the baseband data to generate DBPSK and DQPSK and drives the switch to backscatter the incident tone signal. The baseband data are the select inputs to a 2-bit multiplexer which switches between the four phases of the 12.375 MHz clock to generate the phase modulated data. The multiplexer output is buffered and used to drive the RF switch, which toggles the antenna between open and short impedance state.

Table 1 shows the power consumption of our design at 1 Mbps and 11 Mbps which was computed using the Cadence spectre and Synopsis Design Complier toolkits [5, 19]. Passive Wi-Fi's IC implementation for 1 Mbps and 11 Mbps consumes a total of 14.5 and 59.2 μW of power respectively. The digital frequency synthesizer is clocked for DQPSK and consumes a fixed power for all data rates. The power consumption of the baseband processor that generates the 802.11b packets scales with the data rate and consumes 30% and 80% of total power for 1 and 11 Mbps respectively.

5 Evaluation

5.1 Physical Layer Performance

We first evaluate the range and then the effect of the frequency shift used in our system. Finally, we present results for all four 802.11b bit rates.

5.1.1 RSSI in Line-of-sight scenarios

We run experiments in two line-of-sight scenarios.

Deployment scenario 1. We fix the distance between the passive Wi-Fi device and the plugged-in device. We then

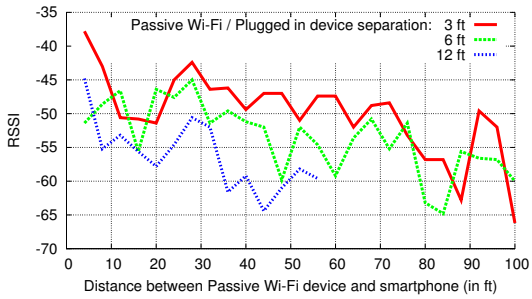


Figure 10: **RSSI in deployment scenario 1.** We move the phone away from the passive Wi-Fi device.

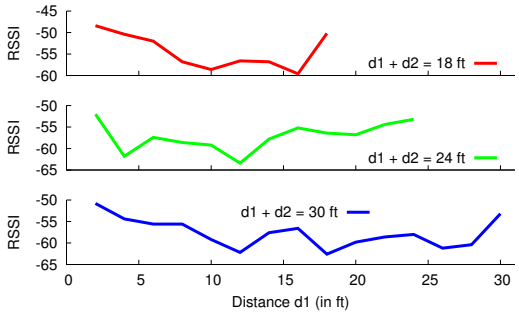


Figure 11: **RSSI in deployment scenario 2.** d_1 (d_2) is the distance between the passive Wi-Fi and plugged-in device (Wi-Fi receiver). The passive Wi-Fi device moves alone the line joining the other two devices.

move the Wi-Fi receiver away from the passive Wi-Fi device and measure the RSSI of the passive Wi-Fi transmissions as seen by the receiver. We run the experiments in the CSE atrium where the maximum distance possible when the passive Wi-Fi device and Wi-Fi receiver were placed on either end was around 100 feet. In our experiments, we set the passive Wi-Fi device to generate 802.11b beacon packets at 1 Mbps. These packets have a payload of 68 bytes where the SSID is set to *WiLab_0000* and are transmitted every 15 ms. We set the plugged-in device to transmit its tone 12.375 MHz from the center of Wi-Fi channel 1 between channel 1 and 6. We use an HTC One (M7) phone as our Wi-Fi receiver. Since the passive Wi-Fi device is transmitting Wi-Fi beacons, it appears as a Wi-Fi AP at the smartphone. To measure the RSSI values of these packets, we use a third party Android app called Wifi Analyzer [3] that provides the RSSI value as shown in Fig. 12.

In each experiment, we hold the smartphone in our hand and measure the reported RSSI values as we walk away from the passive Wi-Fi device. The measurements are taken at increments of 4 feet. Fig. 10 plots the results for three different values of the distance between the passive Wi-Fi transmitter and the plugged-in device. The x-axis plots the distance between the passive Wi-Fi transmitter and the Wi-Fi receiver while the y-axis plots the reported RSSI values. The plots show that as expected, the RSSI values reduce as the phone moves away

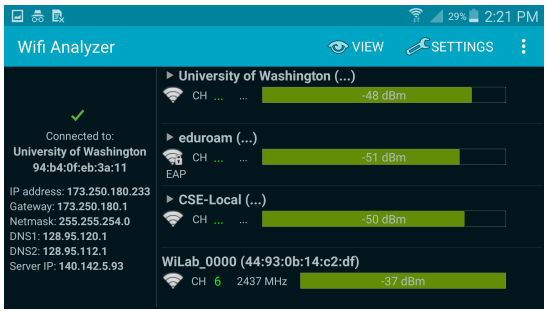


Figure 12: **Snapshot of the Wi-Fi analyzer app.** *WiLab_0000* corresponds to passive Wi-Fi beacons.

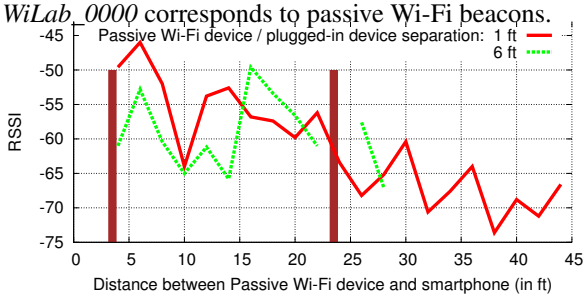


Figure 13: **RSSI in deployment scenario 1 in the presence of walls.** The brown blocks show the wall positions.

from the passive Wi-Fi device. Further, as predicted by our analysis in §2.4, the range of our passive Wi-Fi transmissions reduce with the distance between the passive Wi-Fi transmitter and the plugged-in device. When the separation between the passive Wi-Fi transmitter and the plugged-in device is 3 or 6 feet, the range of the passive Wi-Fi transmissions spans the entire length of the CSE atrium. The range is around 55 feet when this separation is 12 feet. This reduced range is due to a combination of multipath and a weak backscatter signal.

Deployment scenario 2. Next we place the plugged-in device and the Wi-Fi receiver at a distance $d_1 + d_2$. We move the passive Wi-Fi transmitter along the line connecting these two devices. As above the passive Wi-Fi transmitter is set to generate 802.11b beacon packets at 1 Mbps and the plugged-in device transmits its tone at 12.375 MHz from the center of Wi-Fi channel 1. We collect the RSSI values from a HTC One (M7). Fig. 11 plots the results for three different values of the distance between the plugged-in device and the Wi-Fi receiver ($d_1 + d_2$). Each point on the x-axis denotes the distance between the passive Wi-Fi device and the plugged-in device (d_1). The plots show that the RSSI values are the highest when the passive Wi-Fi transmitter is either close to the Wi-Fi receiver or the plugged-in device. Further, the RSSI values are lower at the mid point between the two devices, confirming our theoretical analysis.

5.1.2 RSSI in Through-the-Wall Scenarios

We rerun experiments in the above deployment scenarios but now in the presence of walls. In the first deployment,

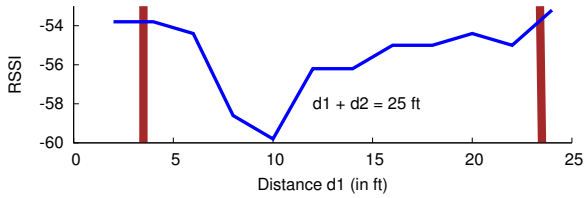


Figure 14: **RSSI in deployment scenario 2 in the presence of walls.** The brown blocks denote the walls. d_1 (d_2) is the distance between the passive Wi-Fi device and plugged-in device (Wi-Fi receiver).

we place the passive Wi-Fi device and the plugged-in devices at distances of 1 and 6 feet from each other. As the Wi-Fi receiver moves away from the passive Wi-Fi device, it is separated by multiple double sheet-rock (plus insulation) walls with a thickness of approximately 5.7 inches. As before, we use an HTC One (M7) phone as our Wi-Fi receiver and set the plugged-in device to transmit with a 12.375 MHz frequency offset from channel 1. The passive Wi-Fi device periodically transmits Wi-Fi beacons at 1 Mbps and we measure the RSSI values as reported by the Wi-Fi receiver. Fig. 13 shows that the range is now around 28 feet when the distance between the passive Wi-Fi device and the plugged-in device is 6 feet. This is expected because the signals get attenuated by two walls before arriving at the Wi-Fi receiver.

In the second deployment, we fix the location of the plugged-in device in the first room and place the Wi-Fi receiver in the third room at a distance of 25 feet. We then move the passive Wi-Fi device along the line connecting the above two devices and measure the RSSI reported by the Wi-Fi receiver. Fig. 14 plots the RSSI results and show that they follow a similar trend as before and work even in the presence of attenuation from walls.

5.1.3 Effect of different frequency shifts

We evaluate how different frequency shift values effect passive Wi-Fi performance. To do this, we place the passive Wi-Fi transmitter and plugged-in device 6 feet from each other. We move a Wi-Fi receiver away from the passive Wi-Fi device in a 50 foot long space. The passive Wi-Fi device transmits 1 Mbps Wi-Fi packets with a payload of 512 bytes on channel 1. We use the Intel 5350 chipset as a Wi-Fi receiver which runs tshark to log all the packets that are successfully decoded by it. The passive Wi-Fi transmitter consecutively transmits 200 unique sequence numbers in a loop using which we compute the packet error rate at the Wi-Fi receiver. We repeat these experiments for three different shifts.

Fig. 15 plots the PER at the Wi-Fi receiver as a function of distance between the passive Wi-Fi transmitter and the Wi-Fi receiver. The figure shows that the PER is consistently around 20% when we use frequency shifts of 44 and 16.5 MHz. For comparison, we measured

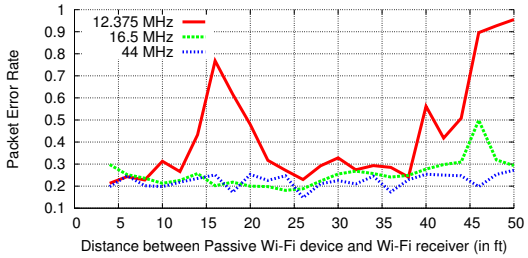


Figure 15: **Effect of different frequency shifts.** The PERs are very stable with 16.5 MHz and 44 MHz offsets.

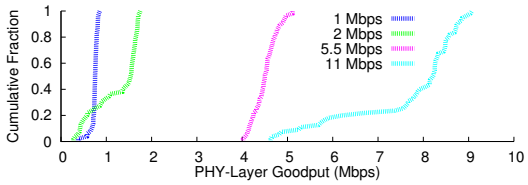
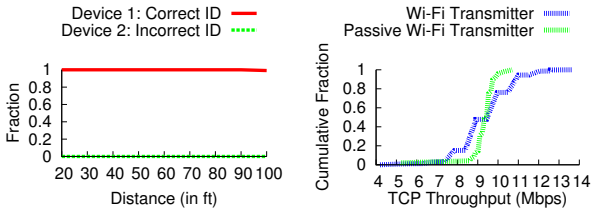


Figure 16: **All 802.11b bit rates.** Our design can generate 802.11b transmissions across all four bit rates.

the PER for a conventional Wi-Fi transmitter placed 10 feet away and observed similar PER values. The interesting observation however is that when the shift is 12.375 MHz, we see a large variation in the PER as the location of the Wi-Fi receiver changes. This is because of two related reasons. First, when the shift is small, the tone from the plugged-in device is very close to the desired Wi-Fi channel. Second, because of multipath, different locations see different signal strength differences between the passive Wi-Fi device and the out-of-band interference from the plugged-in device. When the shift is small, this out-of-band interference can still be significant in certain locations to create losses. We note that while a 44 MHz shift is too high to be within the ISM band, a 16.5 MHz shift has PERs that are stable across locations and yet is small to be within the ISM band while generating packets on all Wi-Fi channels.

5.1.4 Higher 802.11b bit rates

Finally, we show that passive Wi-Fi can generate all 802.11b bit rates. We separate the passive Wi-Fi and plugged-in device by 6 feet. We change the Wi-Fi receiver location to five spots in a 15×24 ft room. The plugged-in device is set to use a 12.375 MHz offset. For each Wi-Fi receiver location, the passive Wi-Fi device transmits 802.11b packets at 1, 2, 5.5 and 11 Mbps. For each bit rate, the passive Wi-Fi device sends 200 packets with a 512 byte payload with different sequence numbers. The Wi-Fi receiver (Intel 5350) is configured to compute the effective PHY goodput achieved by multiplying the transmitted Wi-Fi bit rate with the fraction of packets that are decoded. Fig. 16 plots a CDF of the PHY-layer goodput across the five locations demonstrating that we can generate all four 802.11b bit rates.



(a) ID Detection *(b) Wi-Fi Coexistence*

Figure 17: **Passive Wi-Fi network performance.**

5.2 Passive Wi-Fi Network Performance

As described in §3.1 to coexist in the ISM band, the plugged-in device first performs carrier sense and then signals the passive Wi-Fi device to transmit. In this section, we first evaluate how well the signaling mechanism works. We then describe how our overall carrier sense mechanism works in the presence of other Wi-Fi devices.

5.2.1 Evaluating the signaling mechanism

The plugged-in device transmits a packet with a 10-bit ID that is unique to each passive Wi-Fi device. We evaluate two aspects: (1) the probability with which the signal from the plugged-in device trigger transmissions from the correct passive Wi-Fi device and (2) the probability that it would trigger the wrong passive Wi-Fi device. To evaluate this we consider the worst-case scenario: two devices that have IDs that differ by just one bit. We set the plugged-in device to transmit the signaling packet with the ID of the first device. We move the two passive Wi-Fi devices away from the plugged-in device. At each distance value, the plugged-in device is configured to transmit the signaling packet for a total of 1890 times. The passive Wi-Fi devices use an envelope detector to correlate for their specific ID. We compute the fraction of the 1890 signaling packets that are decoded and match the ID of the passive Wi-Fi device. We run these experiments in the UW CSE atrium for increasing distances from the plugged-in device. Fig. 17(a) show the fraction of signaling packets that match the ID of the two passive Wi-Fi devices as a function of the distance from the plugged-in device. The plot shows that neither device incorrectly decodes the ID. This is because our receiver builds on our prior work [25, 26, 29, 42] and has gone through multiple iterations to improve its reliability.

5.2.2 Evaluating passive Wi-Fi's carrier sense

The plugged-in device performs carrier sense and signals a specific passive Wi-Fi device to transmit. To compare how our mechanism compares to standard Wi-Fi, we compare the performance of a concurrent Wi-Fi transmitter-receiver pair in the presence of a passive Wi-Fi transmitter with that of a traditional Wi-Fi transmitter. We use two Intel 5350 Wi-Fi chipsets to transmit and receive Wi-Fi packets using iperf. The devices use the chipset's default bit rate adaptation. We run experiments in two scenarios: 1) we use a Ralink RT2070 Wi-Fi chipset to transmit packets at 1 Mbps every 15 ms and 2) we set our passive Wi-Fi device to transmit its packet every 15 ms at 1 Mbps using our carrier sense mechanism. We measure the throughput achieved by a concurrent Wi-Fi transmitter-receiver pair in the presence of these two devices. Fig. 17(b) plots the TCP throughput and shows that passive Wi-Fi has a similar impact on the ongoing flow as a traditional Wi-Fi transmitter. This is because, passive Wi-Fi adds only a small fixed 100 μs overhead. This small overhead is however overshadowed by transient changes in network conditions.

5.3 Applications

We first consider low latency sensors like microphones and cameras that transmit continuously. We then analyze duty-cycled sensors.

1) Low power microphones consume 17 μW [1] and an ADC digitizing the microphone output consumes 33 μW [1], resulting in 50 μW for the sensing subsystem. If we use an IoT Wi-Fi chipset by Gainspan or TI to continuously transmit audio, the active Wi-Fi transmitter consumes 670 mW [9, 21]. This results in a total power budget of 670.05 mW which is dominated by the Wi-Fi chipset. However, if we use passive Wi-Fi at 1 Mbps, the power budget drops to 65 μW, i.e., a 1000x reduction.

2) A low power camera like OV7690 operating at VGA resolution and capturing one image per second consumes an average of 10 mW [15]. The camera outputs raw data at 2.45 Mbps which can be transferred wirelessly without power hungry on-board compression. Using an IoT Wi-Fi chipset from Gainspan or TI, brings the total power consumption of the system to 680 mW. If we substitute an active Wi-Fi chipset which consumes 670 mW of power with 11 Mbps passive Wi-Fi, we can improve the battery life of Wi-Fi video camera by at least 50x [9, 21].

3) Duty cycled sensors such as iBeacon [11] and home proximity sensors [16] periodically transmit data using Bluetooth Low Energy and ZigBee protocols respectively. They typically transmit beacons/data packets at a rate of 100 ms to 900 ms and last for 3 months to 3 years respectively on a coin cell battery [11]. If we replace the BLE/ZigBee transmitter which consumes 35 mW [20] in transmit mode with passive Wi-Fi consuming 15 μW, the battery life can be extended well beyond 10 years.

6 Related Work

RFID systems. RFID tags backscatter the signal back to a dedicated 900 MHz RFID reader. The use of backscat-

ter as a general communication mechanism, however, has been limited to RFID systems for two key reasons. First, to decode the weak backscattered signals, the reader eliminates the strong signal from the reader using full-duplex radios [27, 43]. This requires expensive circulators and highly linear analog RF front end at the reader that contributes to its high cost. In contrast, Wi-Fi chipsets do not require the specialized components, can be fully integrated in silicon and hence, are orders of magnitude less expensive. Second, enabling backscatter communication with existing devices requires a complete hardware change to their chipsets and incorporating a dedicated full duplex radio; this is a high bar that has limited the adoption of backscatter beyond RFID.

Wi-Fi and ambient backscatter systems. Since 2013, we have introduced the concepts of ambient and Wi-Fi backscatter [25, 28, 37] where battery-free devices communicate with each other by backscattering ambient signals such as TV and Wi-Fi transmissions. The basic difference between these designs and passive Wi-Fi is that Wi-Fi backscatter systems create an additional narrowband data stream to ride on top of existing Wi-Fi signals. In contrast, passive Wi-Fi aims to use backscatter to generate 802.11b transmissions that can be decoded by billions of existing devices with a Wi-Fi chipsets.

In particular, our prior work on Wi-Fi backscatter [25] demonstrated that existing Wi-Fi chipsets can decode backscattered information from a tag using changes to the per-packet CSI/RSSI values at 1 kbps bitrates and a 2 m range. [38] improved the rate of this communication using a full-duplex radio to cancel the high-power Wi-Fi transmissions from the reader and decode the weak phase-modulated narrowband backscattered signal at the reader. This has allowed them to achieve data rates of up to 5 Mbps at a range of 1 m and 1 Mbps at a range of 5 m. A recent news release [14] claims to achieve 330 Mbps Wi-Fi backscatter communication at 2.5 m using a custom IC that implements a full-duplex radio. The challenge with these full-duplex designs is that they have the same problem as conventional RFID designs— they require a custom full-duplex radio to be incorporated at the receiver and hence the backscattered signals cannot be decoded on any of the existing Wi-Fi devices.

Finally, [23] creates Bluetooth signals using subcarrier modulation to create 370 kHz narrowband 2-FSK signals. Instead, we create 22 MHz DSSS/CCK transmissions using backscatter and enable Wi-Fi transmissions. We also present a network-layer stack design that enables us to operate with existing devices in the ISM band.

Duty-cycled radios. The key idea in these systems is to design a custom low power radio transmitter and use a wakeup receiver to duty cycle the transmitter and reduce the average transceiver power consumption [41]. The power consumption of such transmitters at sub-milliwatt output power is in the order of 100 μW [36, 44] to few mWs [22, 35, 40]. Further, such radios use custom protocols supporting 10-100 kbps data rates that require deployment of special purpose receivers and hardware. In contrast, passive Wi-Fi generates Wi-Fi transmissions at tens of microwatts of power. Given the ubiquity of Wi-Fi, this significantly lowers the bar for adoption. Further, the duty cycle operation is orthogonal to passive Wi-Fi and can be used to further reduce the power consumption of a system employing passive Wi-Fi.

Low power Wi-Fi transceivers. The Wi-Fi industry has designed chipsets for IoT applications including QUALCOMM QCA4002 and QCA4004 [18]. These designs reduce the power consumption by decreasing the transmit power by up to a half when in proximity of another device. They also optimize the power consumption of their sleep mode to be less than 1 mW. Gainspan and TI Wi-Fi chipsets incorporate a 20 μW standby mode and can switch to active mode within tens of milliseconds [9, 21]. However, their active transmission power is around 600 mW [9, 21] which is orders of magnitude higher than passive Wi-Fi. Intel's Moore's radio [7] designs digital versions for RF components such as frequency synthesizers. This reduces the cost and size of the RF chipset rather than its power — a digital Wi-Fi frequency synthesizer consumes 10-50 mW [7, 24] which is similar in power consumption to its analog counterpart.

Finally, recent low power Wi-Fi receiver designs use techniques like dynamic voltage and frequency scaling [30] and compressive sensing [31]. In particular, SloMo [31] leverages the sparsity inherent to 802.11b DSSS signals using compressive sensing to operate the radio at a lower clock rate. Enfold [30] extends this to work with OFDM modulation. Our work on enabling ultra-low power Wi-Fi transmissions is complimentary to this work and can in principle be integrated together.

7 Conclusion

We demonstrate for the first time that one can generate 802.11b transmissions using backscatter communication, while consuming 4-5 orders of magnitude lower power than existing Wi-Fi chipsets. Wi-Fi has traditionally been considered a power-consuming system. Thus, it has not been widely adopting in the sensor network and IoT space where low-power devices primarily transmit data. We believe that, with its orders of magnitude lower power consumption, passive Wi-Fi has the potential to transform the Wi-Fi industry.

Acknowledgements. We thank Ben Ransford and our shepherd Jon Howell. This work was funded in part by NSF under awards CNS-1452494 and CNS-1407583 and a Qualcomm fellowship.

References

[1] ADMP801. http://www.cdiweb.com/datasheets/invensense/ADMP801_2_Page.pdf.

[2] Altera de1 fpga development board. http://www.terasic.com.tw/cgi-bin/page/archive.pl?No=83.

[3] Android wi-fi analyzer. https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer&hl=en.

[4] Atheros targets cellphone with wi-fi chip. http://www.eetimes.com/document.asp?doc_id=1172134.

[5] Cadence rfspectre. http://www.cadence.com/products/rf/spectre_rf_simulation/pages/default.aspx.

[6] Co-existence of wi-fi and bluetooth radios by marvell. http://www.marvell.com/wireless/assets/Marvell-WiFi-Bluetooth-Coexistence.pdf.

[7] Connecting the future: The latest research from intel labs. http://download.intel.com/newsroom/kits/idf/2012_fall/pdfs/IDF2012_Justin_Rattner.pdf.

[8] Dropcam. https://nest.com/camera/meet-nest-cam/?dropcam=true.

[9] Gainspan gs1500m. http://www.alphamicro.net/media/412417/gs1500m_datasheet_rev_1_4.pdf.

[10] Hms190bms8 by hittite microwave devices. https://www.hittite.com/content/documents/data_sheet/hmc190bms8.pdf.

[11] ibeacons. http://beekn.net/2014/04/will-apple-pull-plug-ibeacon-devices/.

[12] Ieee 802.11 standard, 2012. http://standards.ieee.org/getieee802/download/802.11-2012.pdf.

[13] Max2830 by maxim. https://datasheets.maximintegrated.com/en/ds/MAX2830.pdf.

[14] Nasa news release: A wi-fi reflector chip to speed up wearables. http://www.jpl.nasa.gov/news/news.php?feature=4663.

[15] Ovm 7690 camera module. http://www.ovt.com/uploads/parts/OVM7690_PB(1.0)_web.pdf.

[16] Proximity sensors. https://www.ia.omron.com/products/category/sensors/proximity-sensors/.

[17] Qualcomm atheros 9462. http://www.qca.qualcomm.com/wp-content/uploads/2013/11/AR9462.pdf.

[18] Qualcomm qca4002 and qca4004. http://www.eeworld.com.cn/zt/wireless/downloads/QCA4002-4004FIN.pdf.

[19] Synopsis design complier. http://www.synopsys.com/Tools/Implementation/RTLSynthesis/DesignCompiler/Pages/default.aspx.

[20] TI CC2541. http://www.ti.com/lit/ds/symlink/cc2541.pdf.

[21] TI CC3100MOD. http://www.ti.com/lit/ds/symlink/cc3100mod.pdf.

[22] J. Ayers, N. Panitantum, K. Mayaram, and T. S. Fiez. A 2.4 ghz wireless transceiver with 0.95 nj/b link energy for multi-hop battery-freewireless sensor networks. In *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*, pages 29–30. IEEE, 2010.

[23] J. Ensworth and M. Reynolds. Every smart phone is a backscatter reader: Modulated backscatter compatibility with bluetooth 4.0 low energy (ble) devices. In *RFID, 2015 IEEE International Conference on*.

[24] K. Greene. Intel's tiny wi-fi chip could have a big impact. *MIT Technology review*, 2012.

[25] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall. Wi-fi backscatter: Internet connectivity for rf-powered devices. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, 2014.

[26] B. Kellogg, V. Talla, and S. Gollakota. Bringing gesture recognition to all devices. In *Usenix NSDI*, volume 14, 2014.

[27] P. B. Khannur, X. Chen, D. L. Yan, D. Shen, B. Zhao, M. K. Raja, Y. Wu, R. Sindunata, W. G. Yeoh, and R. Singh. A universal uhf rfid reader ic in 0.18-μm cmos technology. *Solid-State Circuits, IEEE Journal of*, 43(5):1146–1155, 2008.

[28] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient backscatter: Wireless communication out of thin air. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, 2013.

[29] V. Liu, V. Talla, and S. Gollakota. Enabling instantaneous feedback with full-duplex backscatter. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 67–78. ACM, 2014.

[30] F. Lu, P. Ling, G. M. Voelker, and A. C. Snoeren. Enfold: downclocking ofdm in wifi. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 129–140. ACM, 2014.

[31] F. Lu, G. M. Voelker, and A. C. Snoeren. Slomo: Downclocking wifi communication. In *NSDI*, pages 255–258, 2013.

[32] J. Manweiler and R. Roy Choudhury. Avoiding the rush hours: Wifi energy management via traffic isolation. In *MobiSys*, 2011.

[33] R. Mittal, A. Kansal, and R. Chandra. Empowering developers to estimate app energy consumption. In *MobiCom*, 2012.

[34] P. Nikitin and K. Rao. Theory and measurement of backscattering from RFID tags. *Antennas and Propagation Magazine, IEEE*, 48(6):212 –218, december 2006.

[35] B. Otis, Y. Chee, R. Lu, N. Pletcher, and J. Rabaey. An ultra-low power mems-based two-channel transceiver for wireless sensor networks. In *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*, pages 20–23. IEEE, 2004.

[36] J. Pandey and B. P. Otis. A sub-100 w mics/ism band transmitter based on injection-locking and frequency multiplication. *Solid-State Circuits, IEEE Journal of*, 46(5):1049–1058, 2011.

[37] A. N. Parks, A. Liu, S. Gollakota, and J. R. Smith. Turbocharging ambient backscatter communication. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, 2014.

[38] D. Pharadia, K. R. Joshi, M. Kotaru, and S. Katti. Backfi: High throughput wifi backscatter. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015.

[39] J. G. Proakis and M. Salehi. Digital communications. 2005. *McGraw-Hill, New York*.

[40] J. Rabaey, J. Ammer, B. Otis, F. Burghardt, Y. Chee, N. Pletcher, M. Sheets, and H. Qin. Ultra-low-power design. *Circuits and Devices Magazine, IEEE*, 22(4):23–29, 2006.

[41] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, and S. Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, 33(7):42–48, 2000.

[42] 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. *Instrumentation and Measurement, IEEE Transactions on*, 57(11):2608–2615, 2008.

[43] C. Ying and Z. Fu-Hong. A system design for uhf rfid reader. In *Communication Technology, 2008. ICCT 2008. 11th IEEE International Conference on*, pages 301–304. IEEE, 2008.

[44] F. Zhang, Y. Zhang, J. Silver, Y. Shakhsheer, M. Nagaraju, A. Klinefelter, J. Pandey, J. Boley, E. Carlson, A. Shrivastava, et al. A batteryless 19μw mics/ism-band energy harvesting body area sensor node soc. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pages 298–300. IEEE, 2012.