WiFi Without Batteries

Pengyu Zhang\textsuperscript{1}, Colleen Josephson\textsuperscript{1}, Dinesh Bharadia\textsuperscript{2}, Kiran Joshi\textsuperscript{1}, Sachin Katti\textsuperscript{1}

\textsuperscript{1}Stanford University, \textsuperscript{2}MIT
WiFi is widely used in wireless embedded systems.

WiFi is one of the dominant radios used by wireless embedded systems.
WiFi — power bottleneck of wireless embedded systems

WiFi radios consume orders of magnitude higher power compared to computation, storage, and sensing.
Applications enabled by such low-power radios

- **Internet-of-Things**: Connecting billions of devices
- **Mobile health**: Retrieving health info at low cost
- **Wireless sensing**: Tracking motion and location
Can we use WiFi without batteries?
Demo: transmitting ECG data with WiFi at 30µWatts
1. Understand sources of WiFi’s high power consumption

2. Key idea — Backscattering WiFi signals

3. Spectrum efficient backscatter

4. Extension to other radios including Bluetooth and ZigBee
Why WiFi transmission consumes lots of power?

- Complicated baseband processing
- Modulate the baseband signal to 2.4GHz
- Amplify the signal before sending over air

Can we enable WiFi transmission at µWatts?
Radio Frequency Identification (RFID)

RFID communicates data at µWatts and does not need battery.

Can we design a backscatter based WiFi system?
HitchHike: enabling backscatter among WiFi devices

HitchHike — use existing WiFi radios to enable backscatter communication

SenSys 2016: HitchHike: Practical Backscatter using Commodity WiFi
HitchHike: enabling backscatter among WiFi devices

How to do backscatter communication with WiFi?

How to do spectrum-efficient backscatter?
802.11b WiFi uses a finite set of codewords to encode data 0 and data 1.
Key observation

We can transform a codeword $i$ to another codeword $j$ by performing simple operations.

The codeword shown is used by 1Mbps 802.11b WiFi TX.
Codeword translation — embed tag info on WiFi packets
Codeword translation — embed tag info on WiFi packets
Codeword translation — embed tag info on WiFi packets

802.11b WiFi Transmitter

Codeword i

802.11b WiFi Receiver

Codeword j

Tag data 0:
- Codeword i → Codeword i
- Codeword j → Codeword j

Tag data 1:
- Codeword i → Codeword j
- Codeword j → Codeword i

Codebook
How does tag do codeword translation at low power?

802.11b WiFi Transmitter

WiFi Transmitter

802.11b WiFi Transmitter

WiFi Receiver

802.11b WiFi Receiver

How does tag introduce the $X - 1$?
How does tag do codeword translation at low power?

\[
\begin{align*}
\text{codeword 0} & = \text{codeword 1} \times -1 \\
\text{codeword 1} & = \text{codeword 0} \times -1
\end{align*}
\]

\(X -1\) means a 180° phase change on the reflected signal.
How does tag do codeword translation at low power?

- 802.11b WiFi Transmitter
- 802.11b WiFi Receiver

Codeword 0 and codeword 1 transmitted by WiFi Transmitter.

Phase shifter — 500 μWatts
How does tag do codeword translation at low power?

802.11b WiFi Transmitter

WiFi Transmitter

802.11b WiFi Transmitter

- codeword 0
- codeword 1

180° phase difference

delay — 1 μWatts for a 5ns delay
1. Understand sources of WiFi’s high power consumption

2. Key idea — Backscattering WiFi signals

3. Spectrum efficient backscatter

4. Extension to other radios including Bluetooth and ZigBee
Signals from the WiFi transmitter is much louder than the reflected signal.
Frequency-shifting the backscattered signal

\[ B(t) = S(t) \times T(t) \]

- **802.11b WiFi Transmitter**
- **802.11b WiFi Receiver**
- **WiFi Transmitter**
- **WiFi Receiver**
- **WiFi signal**
- **backscatter**
Inefficient spectrum usage — double-side band backscatter

\[ B(t) = S(t) \ast T(t) \]
Single-side band backscatter

[Diagram showing the interaction of WiFi signal and backscatter]
How to decode the tag information?

<table>
<thead>
<tr>
<th>Input signal</th>
<th>tag data 0</th>
<th>tag data 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>codeword i</td>
<td>codeword i</td>
<td>codeword j</td>
</tr>
<tr>
<td>codeword j</td>
<td>codeword j</td>
<td>codeword i</td>
</tr>
</tbody>
</table>

backscatter codeword = tag data XOR WiFi codeword
How to decode the tag information?

tag data = backscatter codeword XOR WiFi codeword
Hardware prototype

Source code to ensure reproducibility:
https://github.com/pengyuzhang/HitchHike
Experiment setup

- Intel 5300 WiFi transmitter
- Backscatter tag
- Apple Macbook Pro laptop
Throughput across distances in line-of-sight deployment
Throughput across distances in non-line-of-sight deployment
HitchHike system power consumption breakdown

Power Consumption (µW)

- Oscillator
- Modulator
- Single-side band

22.5
15
7.5
0
Demo: transmitting ECG data with WiFi at 30µWatts
1. Understand sources of WiFi’s high power consumption

2. Key idea — Backscattering WiFi signals

3. Spectrum efficient backscatter

4. Extension to other radios including Bluetooth and ZigBee
Lessons from HitchHike

1. Backscatter is a power-efficient solution for wireless data transfer

2. Codeword translation → backscatter with existing active radios

3. Correct signal modification is the key of successful codeword translation
Three dimensions of a wireless signal

A wireless signal can be presented by:

\[ S(t) = Ae^{j(2\pi ft + \theta)} \]
Codeword translation can be done in three dimensions

Amplitude modification

Frequency modification

Phase modification

\[
\text{codeword } i = Ae^{i (2\pi ft + \theta)}
\]

\[
\text{codeword } j = Ae^{i (2\pi ft + \theta)}
\]
How does Bluetooth encode its information?

Bluetooth Transmitter  →  Bluetooth packets  →  Backscattered packets  →  Bluetooth Receiver

Transmitter

Receiver

**data 0**  →  Frequency $f_0$

**data 1**  →  Frequency $f_1$

**Codeword**

$$S(t) = Ae^{j(2\pi f_0 t + \theta)}$$

$$S(t) = Ae^{j(2\pi f_1 t + \theta)}$$

**Codebook**
How does tag embed its information?

```
S(t) = Ae^{j(2\pi f_0 t + \theta)}
S(t) = Ae^{j(2\pi f_1 t + \theta)}
S(t) = Ae^{j(2\pi f_0 t + \theta)}
S(t) = Ae^{j(2\pi f_1 t + \theta)}
```

tag data 0:
tag data 1:

codebook
How to do codeword translation for Bluetooth?

\[ S(t) = Ae^{j(2\pi f_0 t + \theta)} \]

\[ B(t) = S(t) * T(t) \]

\[ B(t) = Ae^{j(2\pi f_1 t + \theta)} \]
How to do codeword translation for Bluetooth?

\[ S(t) = Ae^{j(2\pi f_0 t + \theta)} \]

\[ T(t) = Ae^{j(2\pi f_1 - f_0 t + \theta)} \]

\[ S(t)T(t) = Ae^{j(2\pi f_0 t + \theta)}Ae^{j(2\pi f_1 - f_0 t + \theta)} = B(t) \]
How to do codeword translation for Bluetooth?

\[ S(t) = Ae^{j(2\pi f_0 t + \theta)} \]

\[ T(t) = Ae^{j(2\pi f_1 - f_0 t + \theta)} \]

\[ B(t) = S(t) \ast T(t) = Ae^{j(2\pi f_1 t + \theta)} \]

Backscattered packets

Backscatter tag

logic

RF harvester
Backscatter with ZigBee and 802.11g/n WiFi

\[ \text{codeword } i = Ae^{j(2\pi f_t + \theta)} \]

\[ \text{codeword } j = Ae^{j(2\pi f_t + \theta)} \]

Phase modification
• The first backscatter system that can do backscatter between WiFi radios
• Spectrum efficient backscatter via single-side band modulation
• 100~300kbps data rate at a distance of 10m~30m
• We are designing MAC layer for HitchHike devices
• Source code to ensure reproducibility: https://github.com/pengyuzhang/HitchHike