Good morning. I am Pan Hu from University of Massachusetts Amherst. In this talk, I will talking about Laissez Faire Backscatter.
As we all know, low power radios like Bluetooth low energy are crucial for a variety of applications, from wearables to internet of things.

But these radios are not as low power as they seem to be. These radios are advertised as extremely low power and this is true if they are heavily duty-cycled, say 1% of the time. But during the process of active transmission and reception, they consume roughly 20mW, which is a non-trivial amount of power. This “active mode power” hurts us when it comes to streaming data continuously from sensors.

This graph shows the power consumption of low power radios versus typical sensors. The y-axis is in log-scale.

If you look at trends in low-power sensors, you will see that many sensors, even high-rate ones like MEMS accelerometers, microphones, ECG, and even cameras can operate lower than a milliwatt, sometimes only several micro-watts in active mode. In contrast, the radio interface consumes thousands of times more power than the sensor itself. In this case communication becomes a major bottleneck for low power operation.
Is there any technology can bring down the energy consumption, fill the gap between radios and sensors?
A promising technology to bridge this gap is backscatter communication. To explain why backscatter can do it, let me analyze why active radio consumes so much.
Active radios are often designed in a symmetric manner where both the transmitter and receiver generate the carrier and the baseband. This means that each end has radio frequency synthesizers, mixers, low noise amplifiers and power amplifiers which contributes to the high power consumption when the radio is on. Adding all these components up, the total power will exceed 20mW.
Adding all these components up, the total power will exceed 20mW.
In contrast, backscatter is designed in an asymmetric manner. Usually the backscatter reader is much more powerful than the tag. Instead of having power-hungry components at the tag side,
Backscatter enables ultra-low power wireless

Backscatter Reader

Baseband

Mixer

Antenna

Amplifier

RF Synthesizer

Backscatter Tag

Antenna

Baseband

Only one transistor

Backscatter tag uses only one transistor to send information to the reader.
How is this possible? Let me explain it. Instead of sending information from the backscatter tag to the reader directly, the reader first send an unmodulated carrier wave to the tag.
At the tag side, instead of generating the RF carrier wave directly which usually consumes more than 20mW, it just modulates information on top of the carrier sent by the reader. This can be done by using a transistor to modulate the status of the antenna, which consumes about 20uW.
Given we have a power-efficient radio, we need to design protocols very carefully so that they retain the power benefits of backscatter. One important design consideration in backscatter is the trade-off between energy and spectrum efficiency.
We illustrate the example with two configurations. The first configuration is that, we can match the Backscatter data rate to the sampling rate of sensors.
And the second configuration is that, we can match the backscatter data rate to the bandwidth available.
In the first configuration, data is being streamed as and when it is generated by the sensor, for example a microphone connected to a backscatter radio. In this case, the baseband clock is matched to the sensor sampling rate, which is good since a lower rate baseband clock consumes less power.
As a consequence, we tend to have smaller data buffer,
and smaller energy buffer.
#1: Match RF bitrate to sensor sampling rate

Low power consumption

Which result in lower power consumption.
The downside of this scheme is that, since bits are transmitted at a relatively low bit rate,
the signal occupies only a small fraction of the spectrum available.
Consuming that we can have more than 20MHz spectrum as specified by FCC, the spectrum utilization is low.
Another configuration is that, we match the backscatter bitrate to the spectrum available. Given we have such a wide spectrum, we will need a high speech clock to transmit the data.
Consequently, we need to have larger data buffer
#2: Match bitrate to available spectrum

and larger energy buffer to support the high speed transmission.
All these changes will result in higher power consumption.
The good thing is that, since the backscatter transmit faster,
It can occupy larger spectrum
#2: Match bitrate to available spectrum

Leading to better spectrum utilization.
Given we have the tradeoff, let's see how it affects the design of multiple access protocols.
Let’s start with TDMA.
TDMA - Sacrifice Spectrum Efficiency

TDMA divides time into different slots
and each tag choose a time slot to transmit. Here, slower operation is clearly inefficient since slow tags will need large slots to transmit their bits. This will hurt other tags that can transmit faster.
All these result in low spectrum efficiency. However, since each tag is transmit slowly, its energy efficiency is generally good.
An alternative is CDMA.
In CDMA, each bit will be expanded into multiple bits with a pseudo random code.
CDMA - Sacrifice Power Efficiency

Tag1

Tag2

Pseudo Random Code → XOR

Tag1 Encoded

Tag2 Encoded
This has the drawback that the tag has to toggle the transistor many times to transmit one bit of information, and this leads to at least an order of magnitude reduction in power efficiency, although let multiple tags transmitting at the same time can improve the spectrum efficiency.
Another concurrent backscatter protocol is Buzz, which was presented a few years ago at SIGCOMM.
Buzz requires all the tags to toggle their transistors at the same time.
Buzz has better spectrum efficiency than TDMA but incurs the control overhead associated with synchronizing tags.
So, there is a subtle tradeoff between power and bandwidth.
The question is that, can we get both energy and spectrum efficiency?
The answer is LF-Backscatter. A tag may start transmission at any time with any bitrate, therefore the term Laissez faire.
In LF-Backscatter, bits collision happens all the time.
But how can we solve this problem? Take a closer look we can find that the edges in the collided signal still contains information from each tag.
We can assign the edges to each of the tag
and do this repeatedly.
LF-Backscatter - Transmit Whenever It Wants To

Tag1
Tag2
Collided Signal
Signal Edges
Tag1 Edges
Tag2 Edges

Reader
By simply connecting these edges we can recover the signal transmitted by each tag. From a power perspective, this is fantastic since each node can use the minimal rate to operate at its lowest power point. From a spectrum utilization perspective, this is great since we are interleaving transmissions and using the spectrum more efficiently. But the question is whether this picture is realistic.
Here we made two assumptions: the first one is, edges are detectable. The second one is, edges are staggered in time.
Let me explain them one by one. So why are we able to detect signal edges?
In typical active radio communication, this would not be possible since a sharp edge occupies a wide spectrum, which would lead to interference across channels. So, active radios use filters to smooth the edge and limit the bandwidth, which reduces the sharpness of the signal edges. The fact that we have sharp edges means that they should be easier to detect at the reader.
Active radios

Transmitter Baseband

Transmitted Envelop

Edges are not preserved

which reduces the sharpness of the signal edges, making edge detection impossible.
However, there is no such filtering in backscatter.
#1: Why are we able to detect signal edges?

So the edges are clearly detectable.
Let's now look at the receiver side. The receiver also should have the capability to detect signal edges.
In an active radio, the receiver typically sampling at the rate which is usually similar as the transmission bit rate.
The low sampling rate leads to smoothed edges in recovered signal.
However, backscatter is designed in asymmetric manner. The reader is usually much more powerful than the tag, which is capable of sampling at tens or even hundreds of the transmission bitrate.
In this case, there are many samples before, on top of, and after the edge. So the edges can be detected.
Our second assumption is that, edges are staggered.
Why is this true? This actually occurs quite naturally because of the way a backscatter receiver works. A backscatter receiver is a simple envelope detector.
It detected the presence of the reader when the capacitor charges to a certain threshold, after which the comparator generates an output and the tags start to transmit.
However, tags at different locations can have different signal strength, affecting the capacitor’s charging speed. In addition to that, there are about 20% variation in terms of capacitance when manufacturing the capacitors. All these factors will result in different charging curves which result in different start time. As a result, the edges are naturally staggered.
We've seen is that in theory, we can detect signal edges, but is this method sufficiently robust in practice?
Edge detection can be traditionally done by looking at the changes of signal amplitude. But this approach may fail in backscatter. The amplitude can stay the same regardless of the bits transmitted.
Why is this the case? Let’s look at the IQ signal plot when the tag is transmitting 1s. Firstly we will have a strong background signal due to self-interference and environment reflection. Also, we have the signal transmitted by the tag, as shown in blue. The received signal is a combination of both.
Similarly, we can have the IQ plot when the tag is transmitting 0s. But what will happen if the signal transmitted by the tag is orthogonal to the background?
The result will be that there is no change in amplitude regardless of the bits transmitted.
Robust edge detection using IQ vector

How can we address this problem? Instead of looking at the signal edges in the time domain, we should look at the signal edges in in-phase and quadrature dimensions, as shown in red in the example.
How to deal with edge collisions?

So far, I've assumed that signal edges DO NOT collide with each other, but edges can collide especially when there is a large number of tags. One approach is retransmissions, but in many cases, we find that this may not be necessary.
Let us look at this example where the two signal edges collides.
Assuming that two tags have different signal edges $e_1$ and $e_2$. When the two signal edges collide, they add on top of each other. The addition of the two signal edges lead to nine clusters in the IQ dimension where each cluster encodes the state of both edges.
For example, the top cluster is contributed by two rising edges.
and the bottom cluster is caused by falling edges.
How to deal with edge collisions?

The nine clusters are separable if SNR is sufficient. For a colliding edge signal, we only need to classify which cluster it belongs to for decoding.
This approach will not work if there are more than two tags that collide since there are way too many clusters. In this case, we turn to retransmission.
Let's put everything together and provide a complete picture of LF-Backscatter. LF-Backscatter starts by detecting signal edges in IQ domain,
then detect collisions and resolve collision with IQ cluster information.
After that, LF Backscatter assign edges to each tag and decode the data.

I have omitted several low-level details in this talk such as how to associate the edge streams to nodes, how to deal with unknown numbers of tags, unknown bitrates, clock drift, and edge detection errors using a viterbi decoder, and refer you to the paper for these details.
We implemented LF-Backscatter on a software defined radio, USRP N210, and 16 UMass Moo tags. Each tag is able to transmit up to 1 Megabits per second. The front-end of the software defined radio is a SBX daughterboard with a bandwidth of 40MHz. It uses two separated antennas for transmission and reception.
Our experiment shows that LF-Backscatter is able to achieve 15x throughput improvement over TDMA, and 7x over Buzz.
We can observe that LF-Backscatter has stable bits per joule even when the number of tags increases. In contrast, the energy efficiency of TDMA and Buzz decreases when the number of tags increases because there is too much control message overhead for slot messages, repeated transmissions and dealing with collisions. At the same time, LF Backscatter can be 20x more energy-efficient than TDMA/Buzz.
As with every protocol, LF-Backscatter sacrifices something to get the energy and bandwidth benefits.
What does LF-Backscatter sacrifice?

This graph shows the SNR vs Bit error rate of LF-Backscatter versus ASK.

The main downside to LF-Backscatter is that it primarily operates in ranges where SNR is high because those are the conditions under which edges are more robustly decoded. This figure shows the bit error rate of LF-Backscatter across different SNR scenarios. Our experimental results show that LF-Backscatter requires 4dB additional SNR to have the same performance as ASK. This means that the working range of LF-Backscatter will be about 20% shorter than ASK.
Our experimental results show that LF-Backscatter requires 4dB additional SNR to have the same performance as ASK. This means that the working range of LF-Backscatter will be about 20% shorter than ASK.
But one major advantage of LF-backscatter is that it uses a subset of the hardware that is used by a standard passive RFID tag. In other word, it is quite straightforward to switch from using LF-backscatter when conditions are good to and ASK when the SNR is lower. This means that existing tags can, with no hardware modification, use LF-backscatter.
To conclude, LF-Backscatter is a protocol that allows multiple tags to transmit at whatever bitrate they want. At the heart of our work is the energy–bandwidth tradeoff inherent in ultra low-power backscatter-based devices. LF-Backscatter tries to get the best of both worlds by interleaving transmissions to obtain more throughput while allowing them individually to operate at a slow rate to reduce power consumption. The results are dramatically reduced power consumption and increased throughput under moderately high SNR conditions.
In ongoing work, we are building on these ideas to design next-generation backscatter based wearable devices. With that I'll end my talk and I'm happy to take questions.
To conclude, LF-Backscatter is a protocol that allows multiple tags to transmit at whatever bit rate they want to. At the heart of our work is the power–bandwidth tradeoff inherent in ultra-low-power backscatter-based devices. LF-Backscatter looks at the trade off between throughput, power and SNR and finds a good place for operating these ultra-low power tags. In ongoing work, we are building on these ideas to design next-generation backscatter based wearable devices. With that I’ll end my talk and I’m happy to take questions.
To conclude, LF-Backscatter is a protocol that allows multiple tags to transmit at whatever bitrate they want to. At the heart of our work is the power-bandwidth tradeoff inherent in ultra low-power backscatter-based devices. LF-Backscatter looks at the trade off between throughput, power and SNR and finds a good place for operating these ultra-low power tags. In ongoing work, we are building on these ideas to design next-generation backscatter based wearable devices. With that I’ll end my talk and I’m happy to take questions.
To conclude, LF-Backscatter is a protocol that allows multiple tags to transmit at whatever bitrate they want to. At the heart of our work is the power-bandwidth tradeoff inherent in ultra low-power backscatter-based devices. LF-Backscatter looks at the tradeoff between throughput, power and SNR and finds a good place for operating these ultra-low power tags. In ongoing work, we are building on these ideas to design next-generation backscatter based wearable devices. With that I’ll end my talk and I’m happy to take questions.
To conclude, LF-Backscatter is a protocol that allows multiple tags to transmit at whatever bitrate they want to. At the heart of our work is the power–bandwidth tradeoff inherent in ultra low–power backscatter–based devices. LF-Backscatter looks at the trade off between throughput, power and SNR and finds a good place for operating these ultra–low power tags. In ongoing work, we are building on these ideas to design next–generation backscatter based wearable devices. With that I’ll end my talk and I’m happy to take questions.