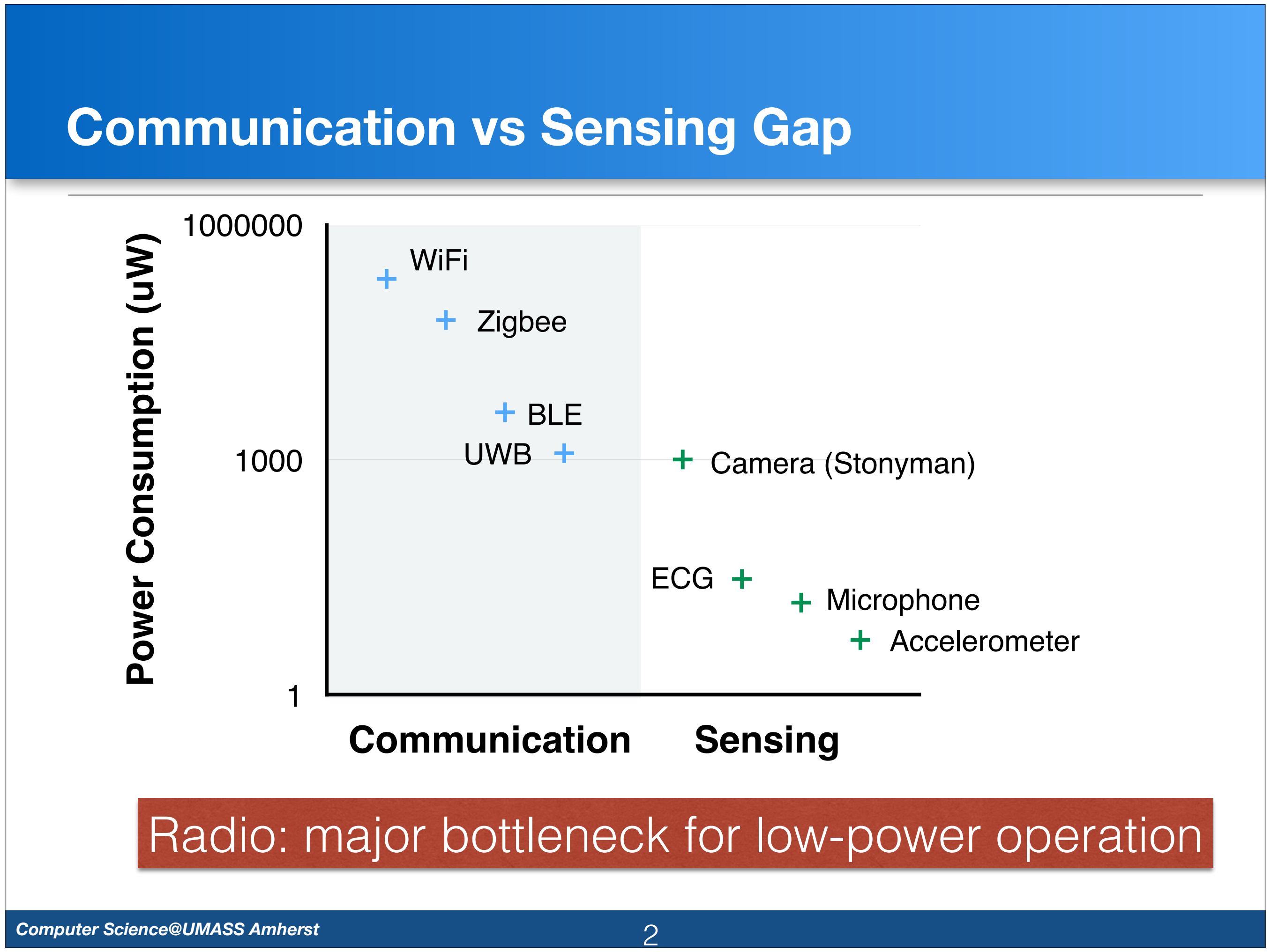
Laissez-Faire: Fully Asymmetric Backscatter Communication

by **Pan Hu**, Pengyu Zhang, Deepak Ganesan
University of Massachusetts Amherst

Good morning. I am Pan Hu from University of Massachusetts Amherst. In this talk, I will talking about Laissez Faire Backscatter.



Communication vs Sensing Gap

Radio: major bottleneck for low-power operation

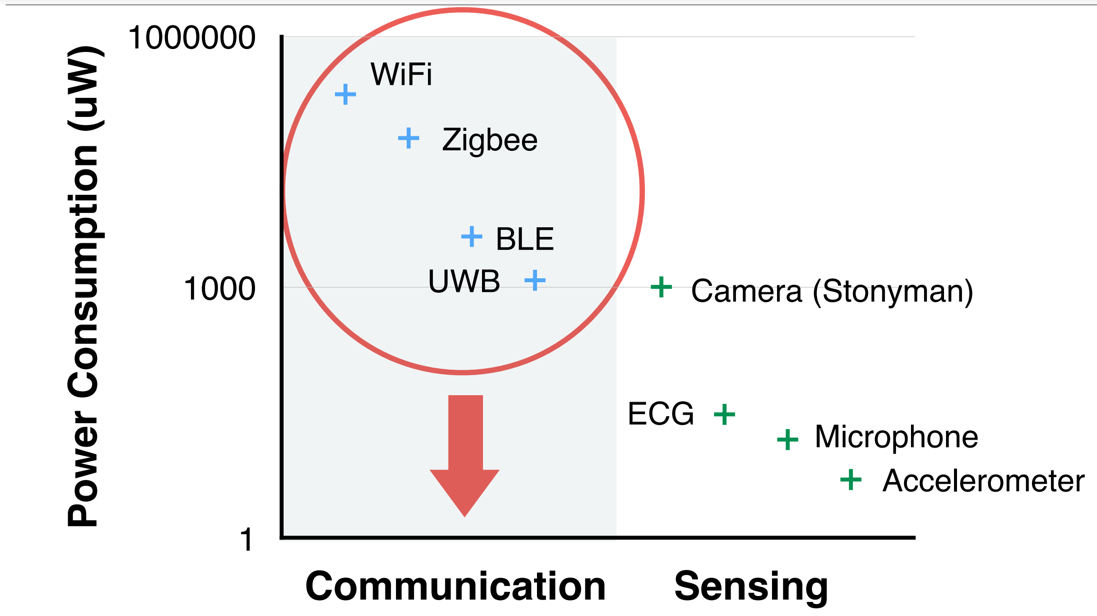
As we all know, low power radios like Bluetooth low energy are crucial for a variety of applications, from wearables to internet of things.

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.

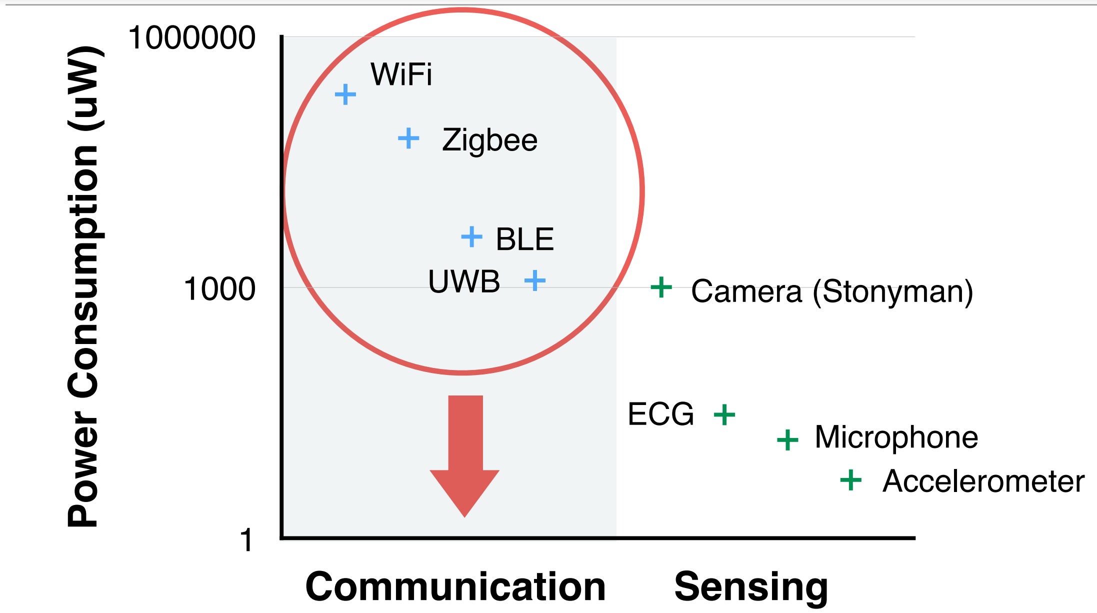
If you look at trends in low-power sensors, you will see that many sensors, even high-rate ones like MEMS accelerometers, microphones, ECG, and even cameras can operate lower than a milliwatt, sometimes only several micro-watts in active mode. In contrast, the radio interface consumes thousands of times more power than the sensor itself. In this case communication becomes a major bottleneck for low power operation.

A BIG Gap Between Radios and Sensors



Is there any technology can can bring down the energy consumption, fill the gap between radios and sensors?

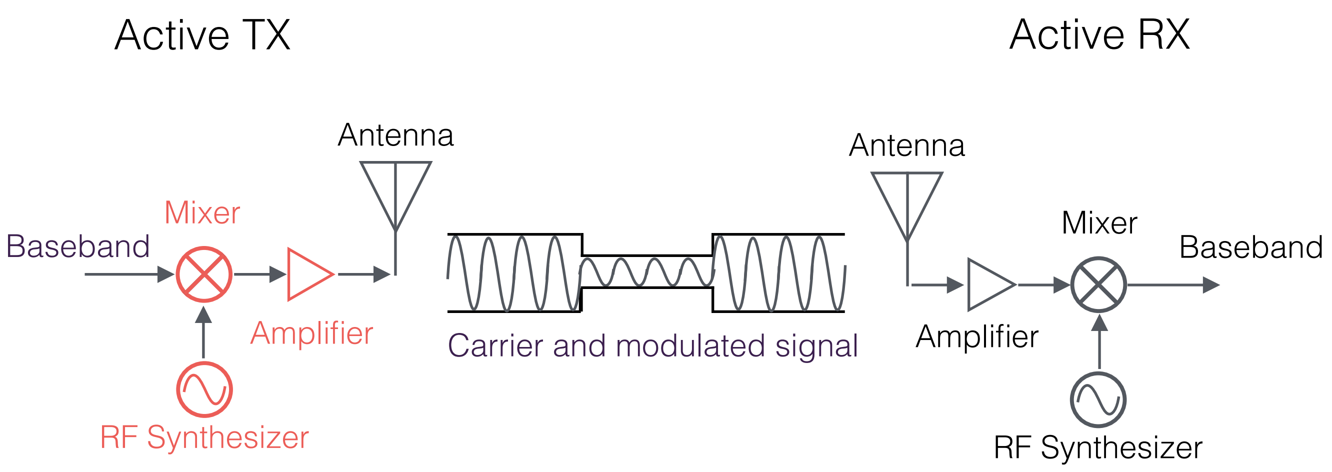
A BIG Gap Between Radios and Sensors



Backscatter can bridge the gap!

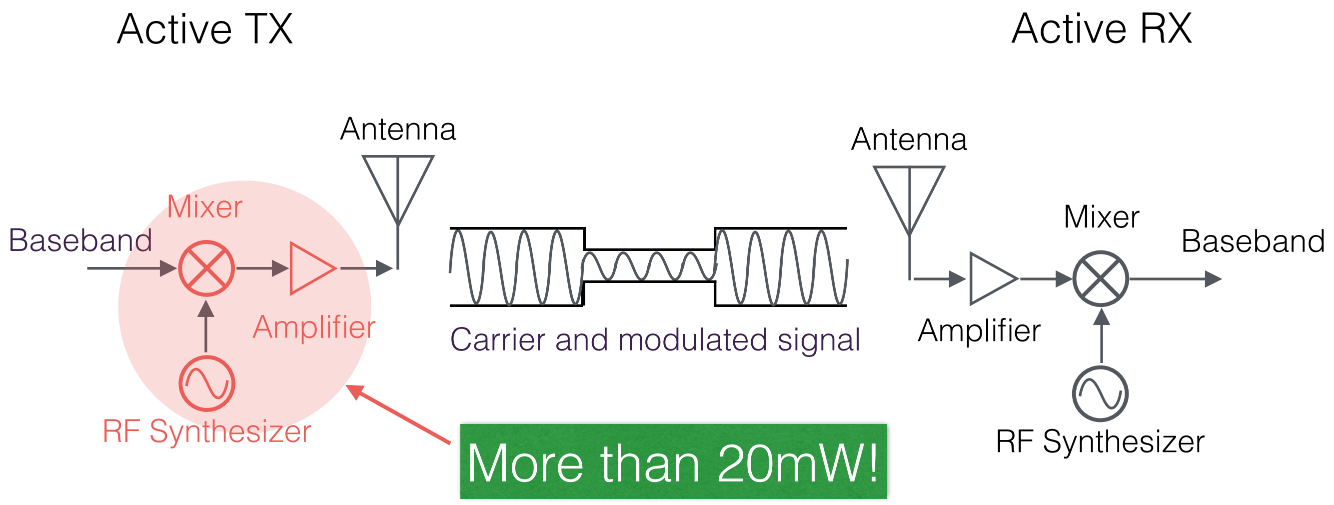
A promising technology to bridge this gap is backscatter communication. To explain why backscatter can do it, let me analysis why active radio consumes so much.

Why are active radios power-hungry?



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.

Why are active radios power-hungry?



More than 20mW!

Adding all these components up, the total power will exceed 20mW.

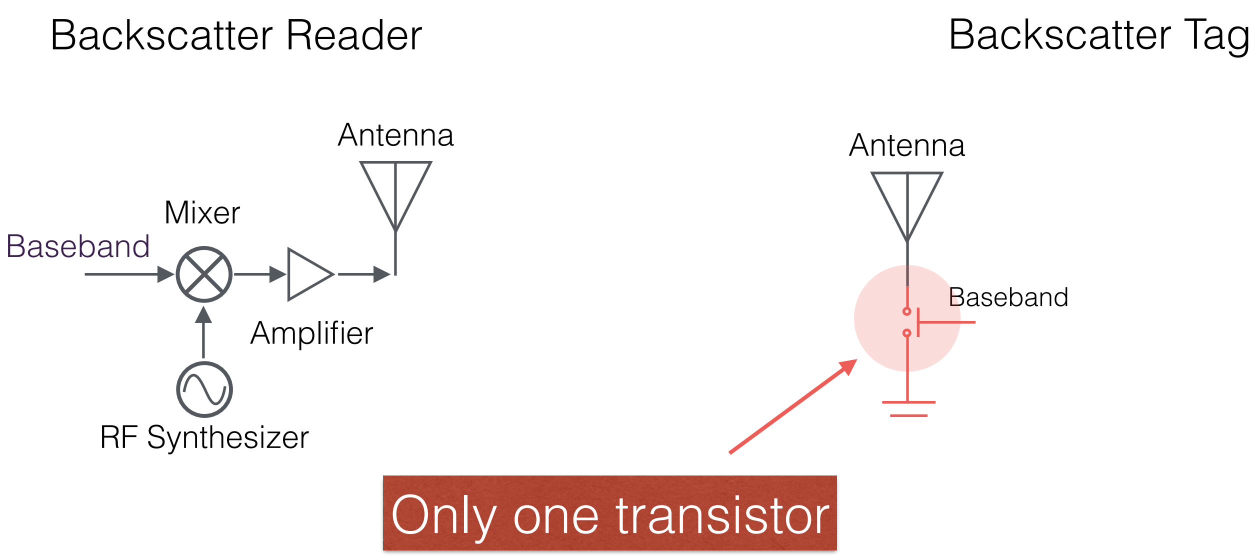
Backscatter enables ultra-low power wireless

Backscatter Reader

Backscatter Tag

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,

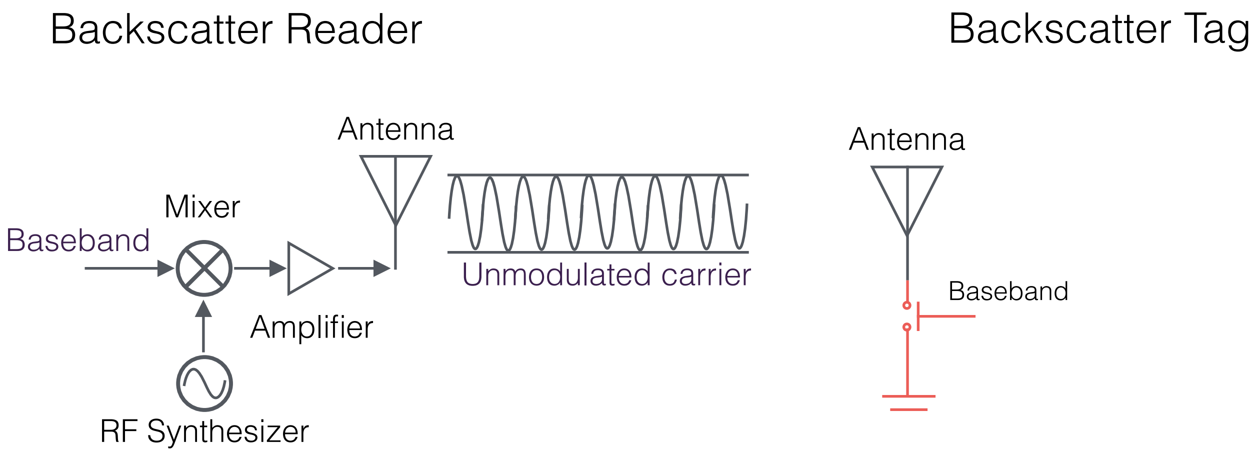
Backscatter enables ultra-low power wireless



Only one transistor

Backscatter tag uses only one transistor to send information to the reader.

Backscatter enables ultra-low power wireless

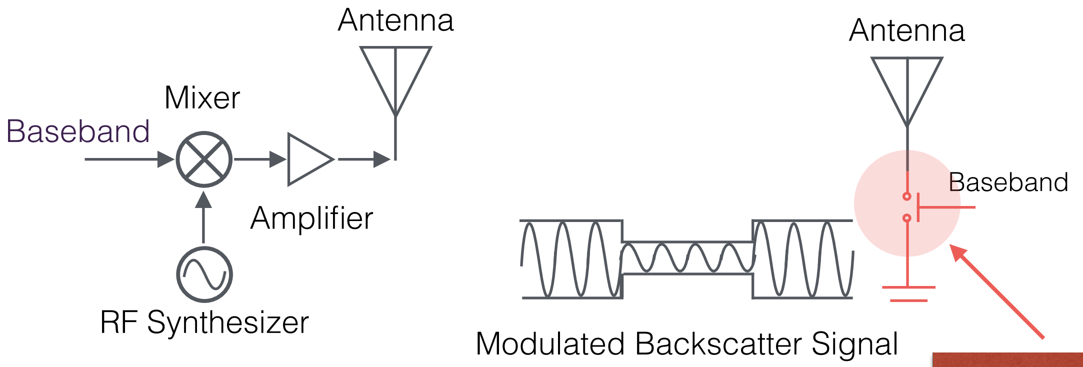


How is this possible? Let me explain it. Instead of sending information from the backscatter tag to the reader directly, the reader first send an unmodulated carrier wave to the tag.

Backscatter enables ultra-low power wireless

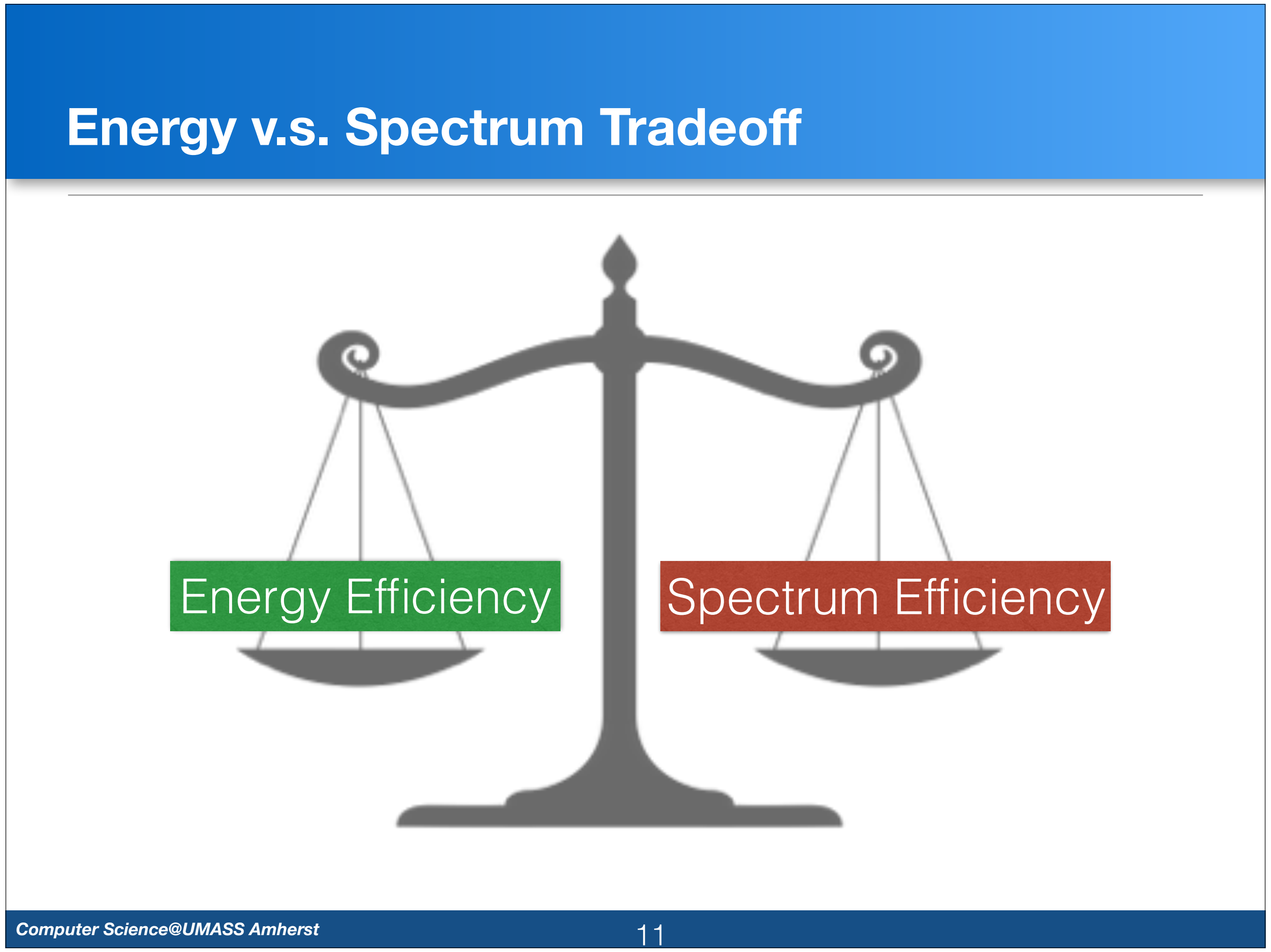
Backscatter Reader

Backscatter Tag



20uW to toggle one transistor.

At the tag side, instead of generating the RF carrier wave directly which usually consumes more than 20mW, it just modulation 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.



Energy v.s. Spectrum Tradeoff

Energy Efficiency

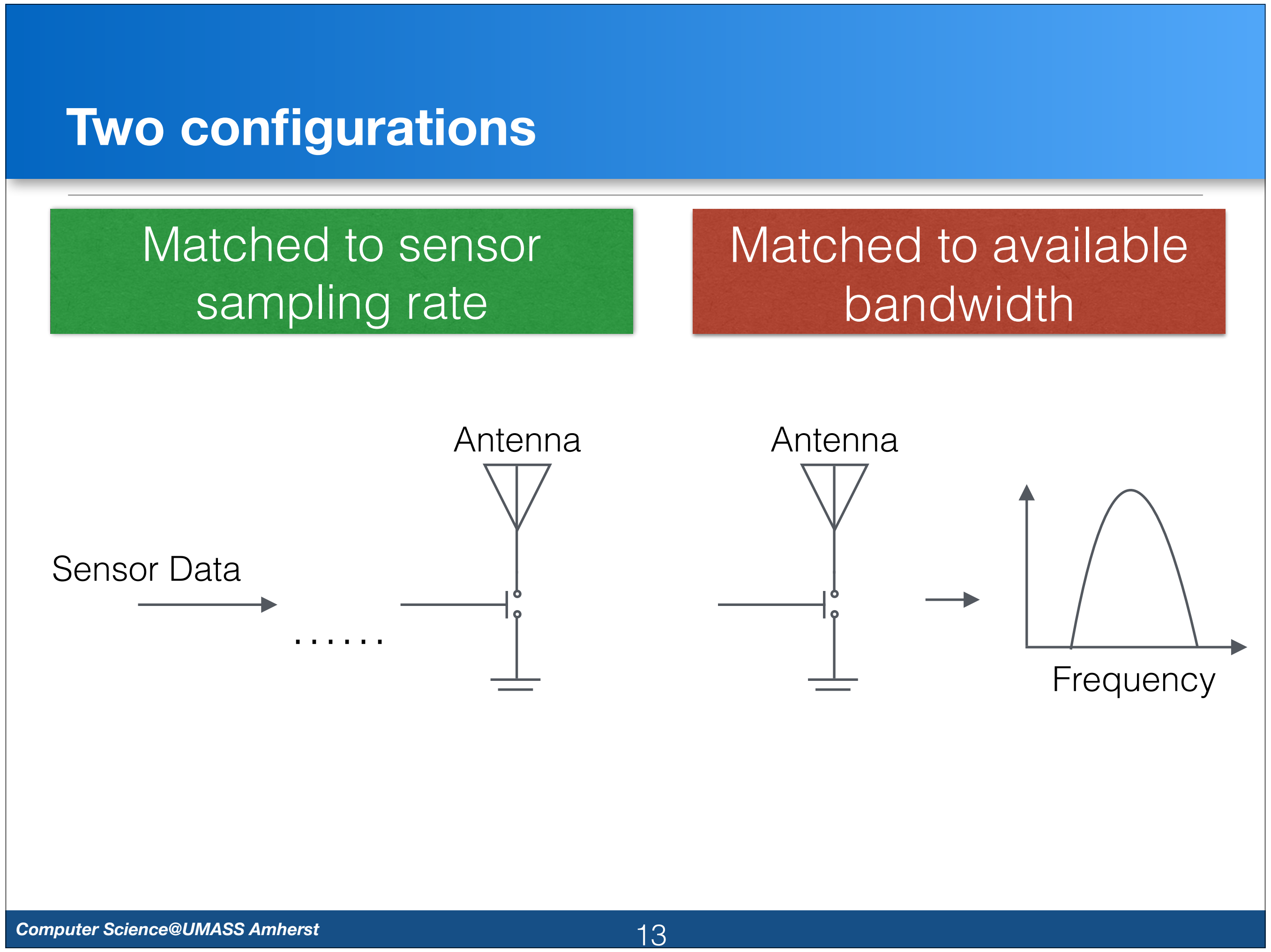
Spectrum Efficiency

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.

Two configurations

Matched to sensor sampling rate

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.



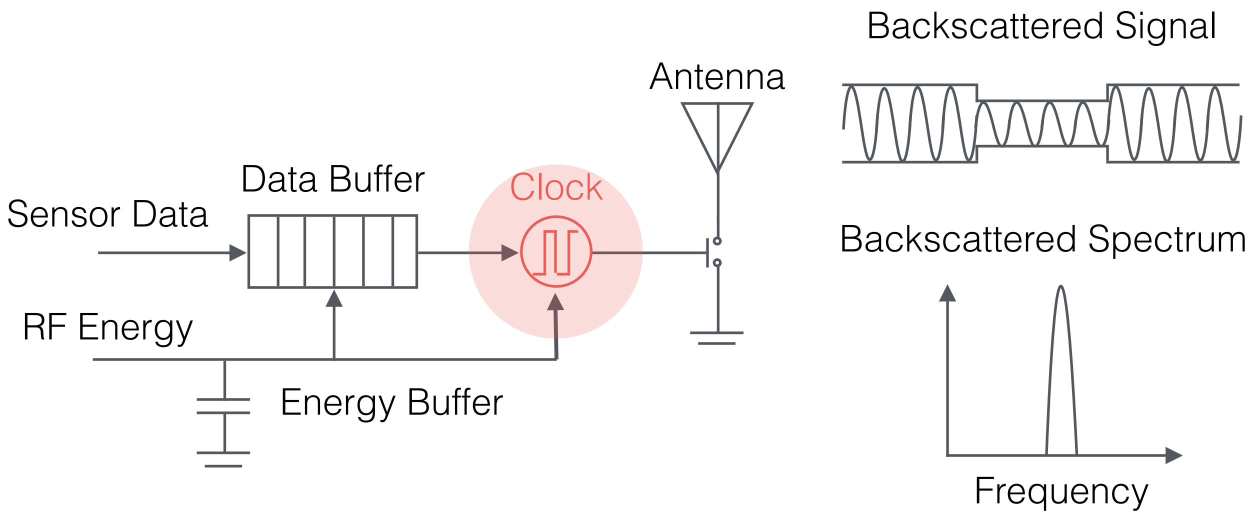
Two configurations

Matched to sensor sampling rate

Matched to available bandwidth

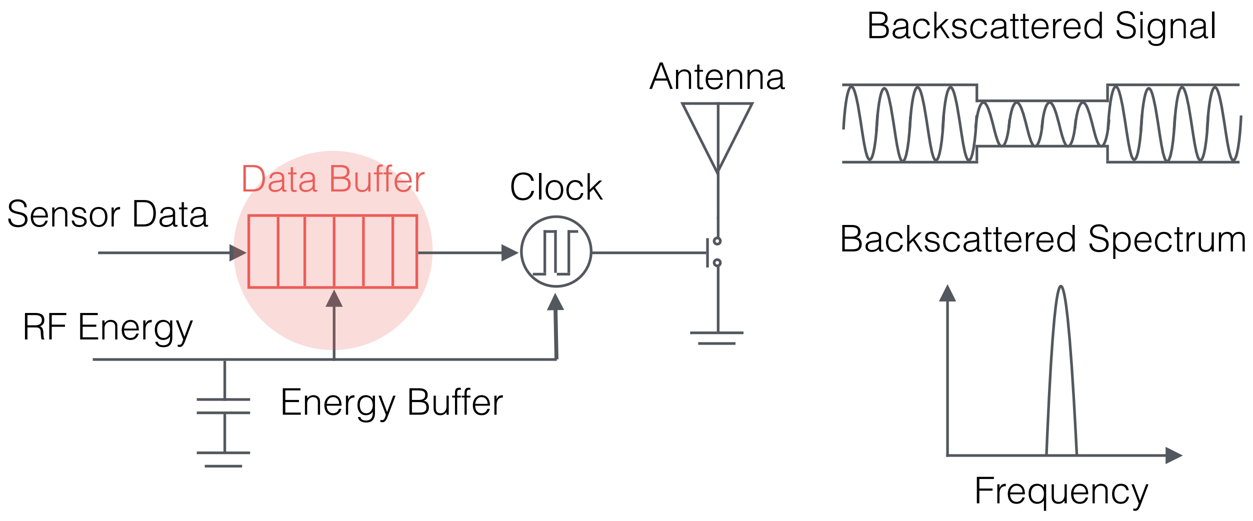
And the second configuration is that, we can match the backscatter data rate to the bandwidth available.

#1: Match RF bitrate to sensor sampling rate



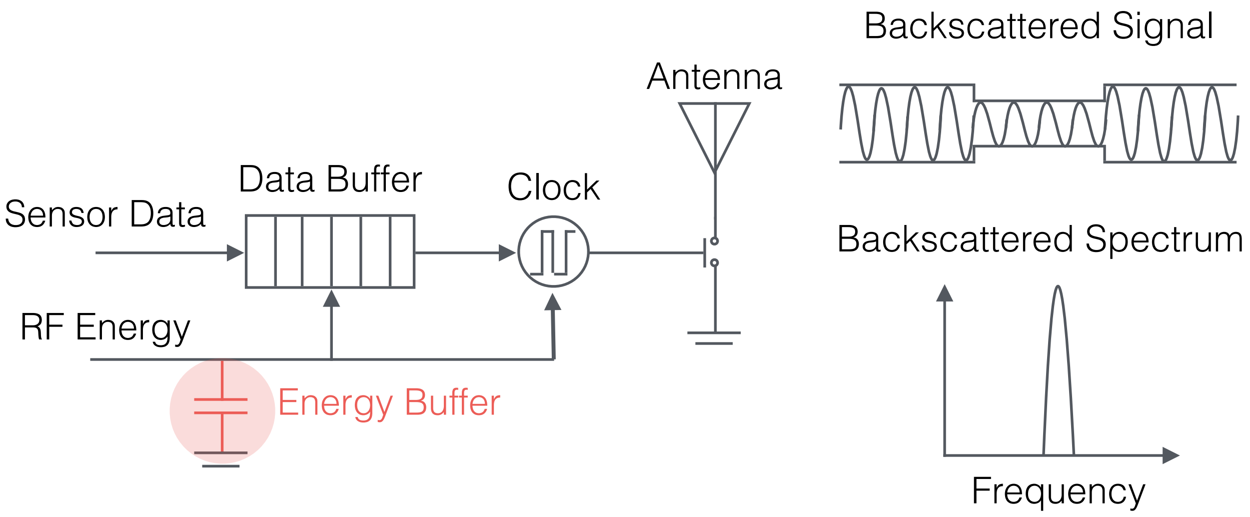
In the first configuration, data is being streamed as and when it is generated by the sensor, for example a microphone connected to a backscatter radio. In this case, the baseband clock is matched to the sensor **sampling** rate, which is good since a lower rate baseband clock consumes less power.

#1: Match RF bitrate to sensor sampling rate



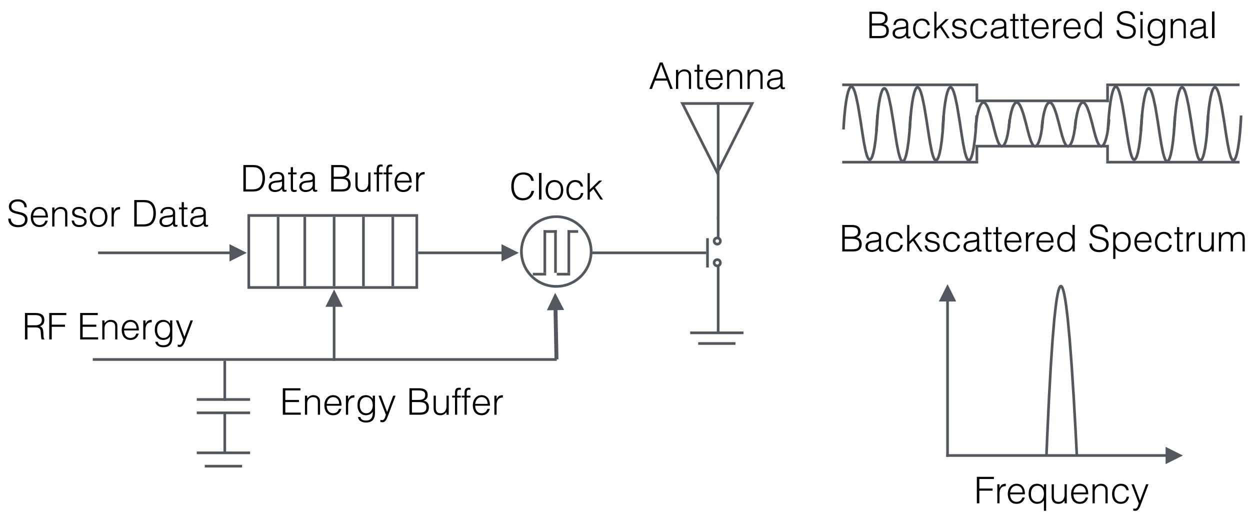
As a consequence, we tend to have smaller data buffer,

#1: Match RF bitrate to sensor sampling rate



and smaller energy buffer.

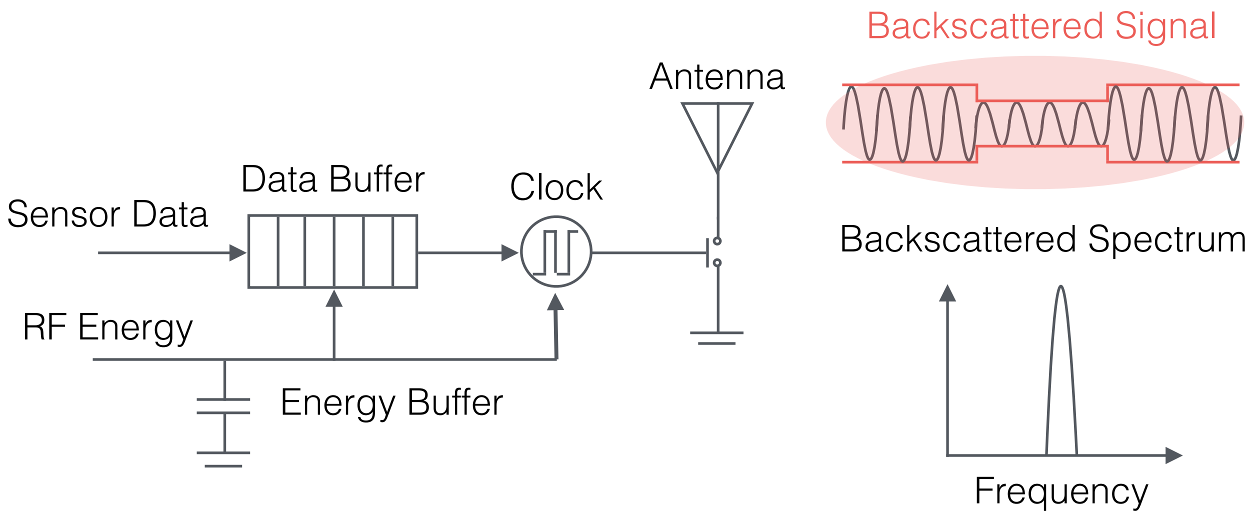
#1: Match RF bitrate to sensor sampling rate



Low power consumption

Which result in lower power consumption.

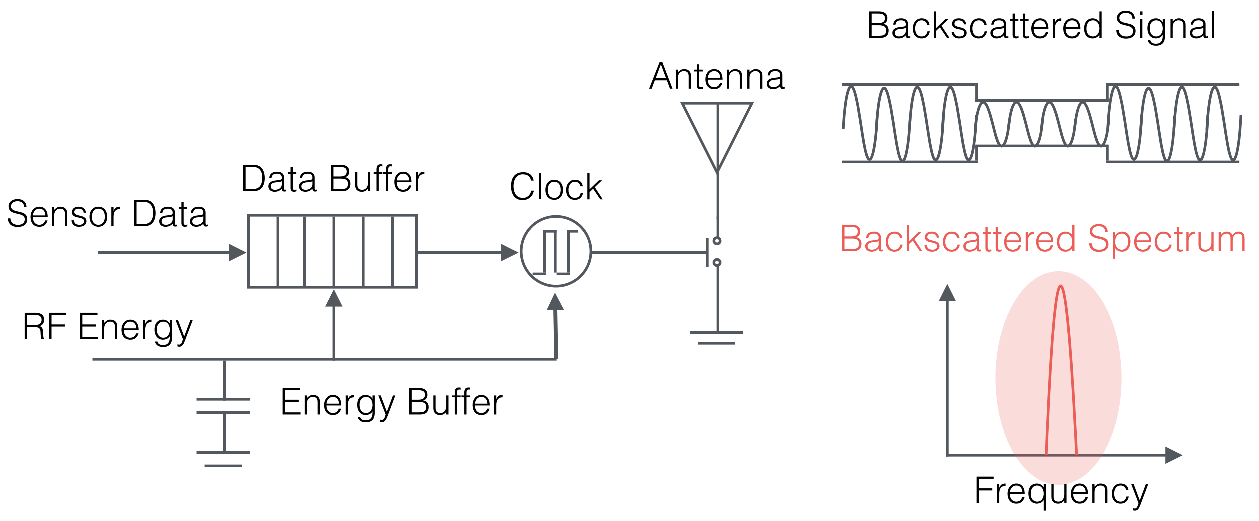
#1: Match RF bitrate to sensor sampling rate



Low power consumption

The downside of this scheme is that, since bits are transmitted at a relatively low bit rate,

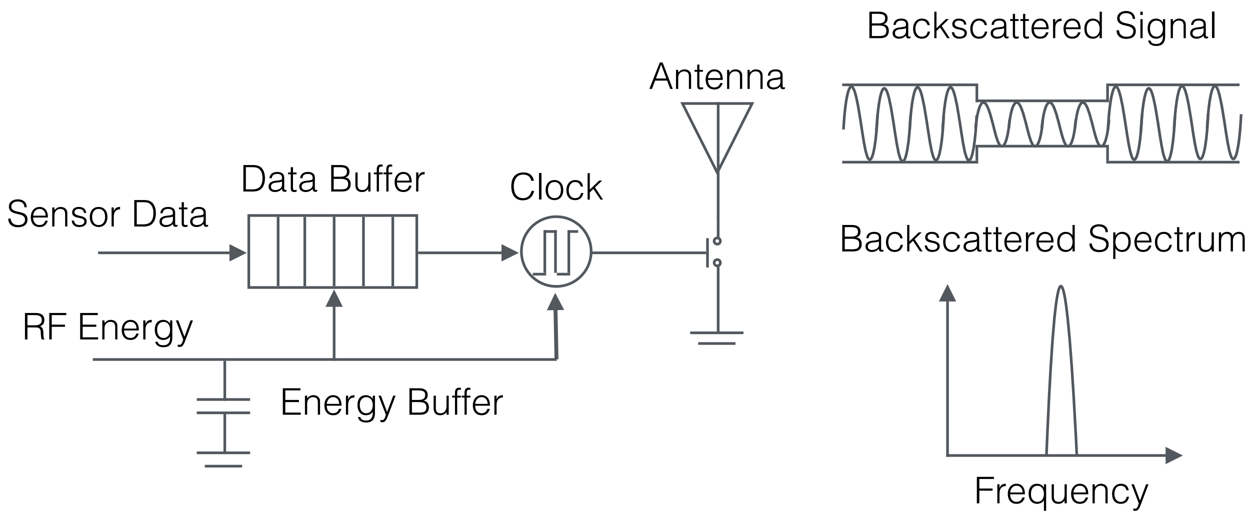
#1: Match RF bitrate to sensor sampling rate



Low power consumption

the signal occupies only a small fraction of the spectrum available.

#1: Match RF bitrate to sensor sampling rate

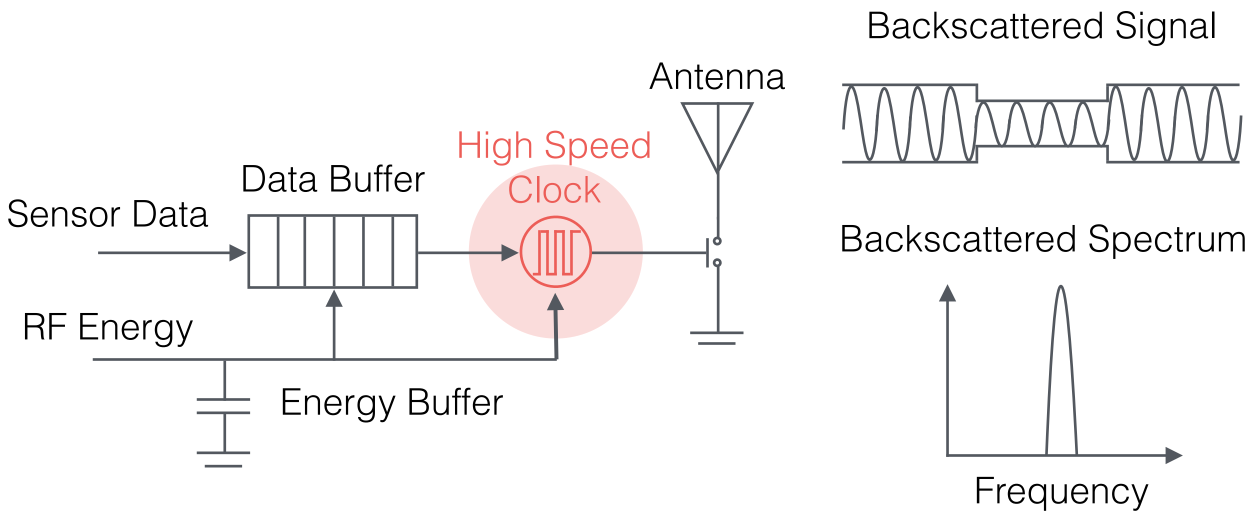


Low power consumption

Low spectrum utilization

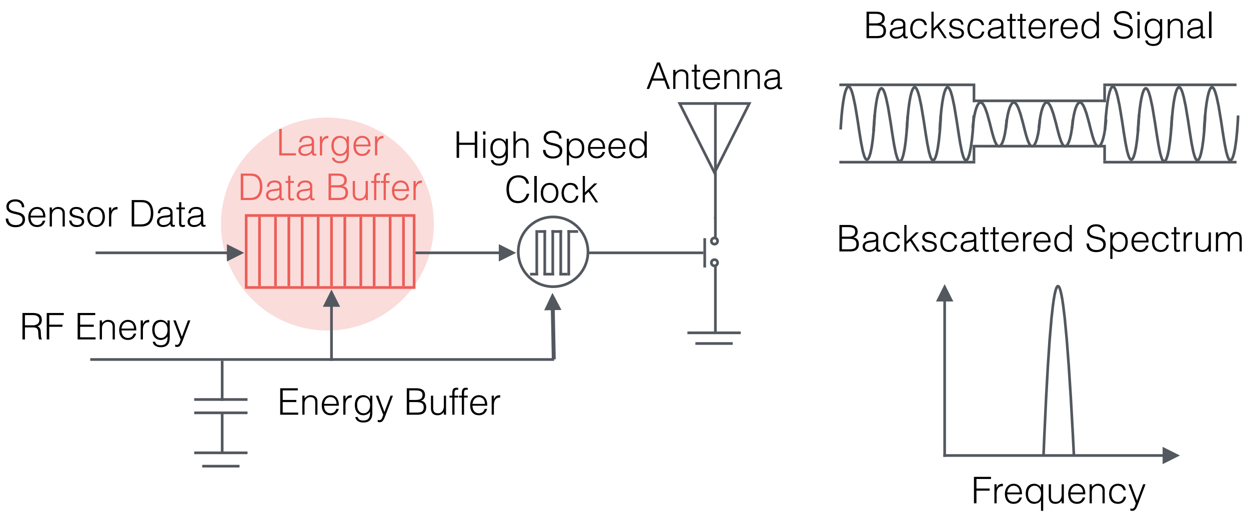
Consuming that we can have more than 20MHz spectrum as specified by FCC, the spectrum utilization is low.

#2: Match bitrate to available spectrum



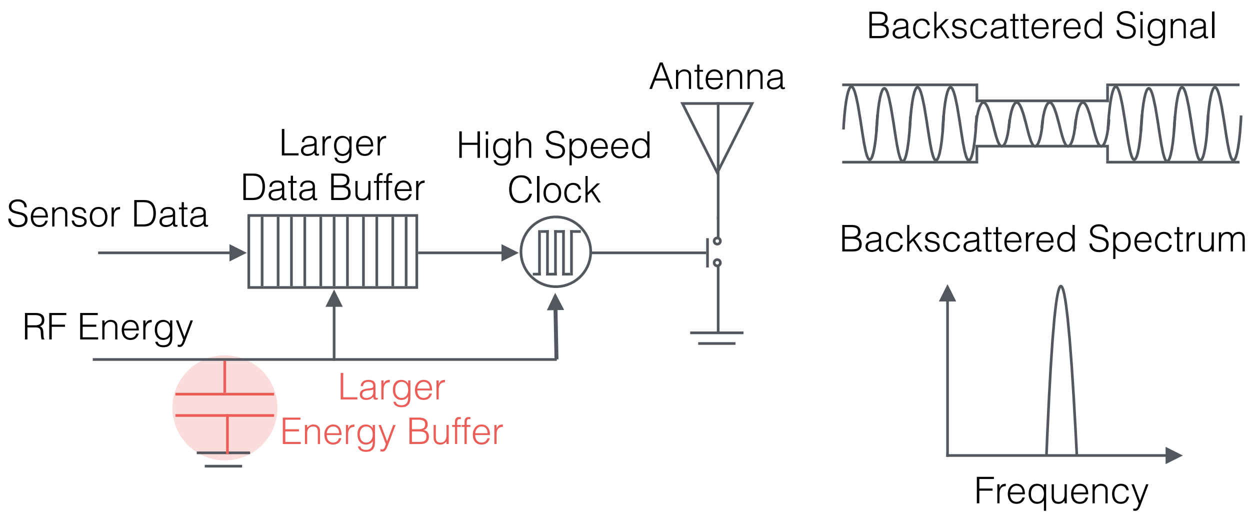
Another configuration is that, we match the backscatter bitrate to the spectrum available. Given we have such a wide spectrum, we will need a high speech clock to transmit the data.

#2: Match bitrate to available spectrum



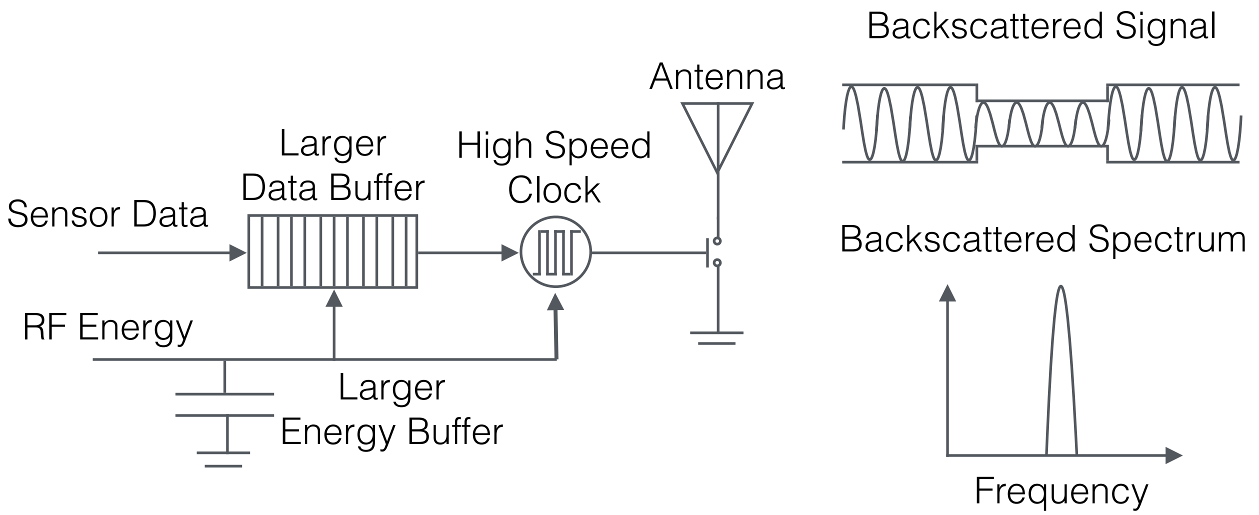
Consequently, we need to have larger data buffer

#2: Match bitrate to available spectrum



and larger energy buffer to support the high speed transmission.

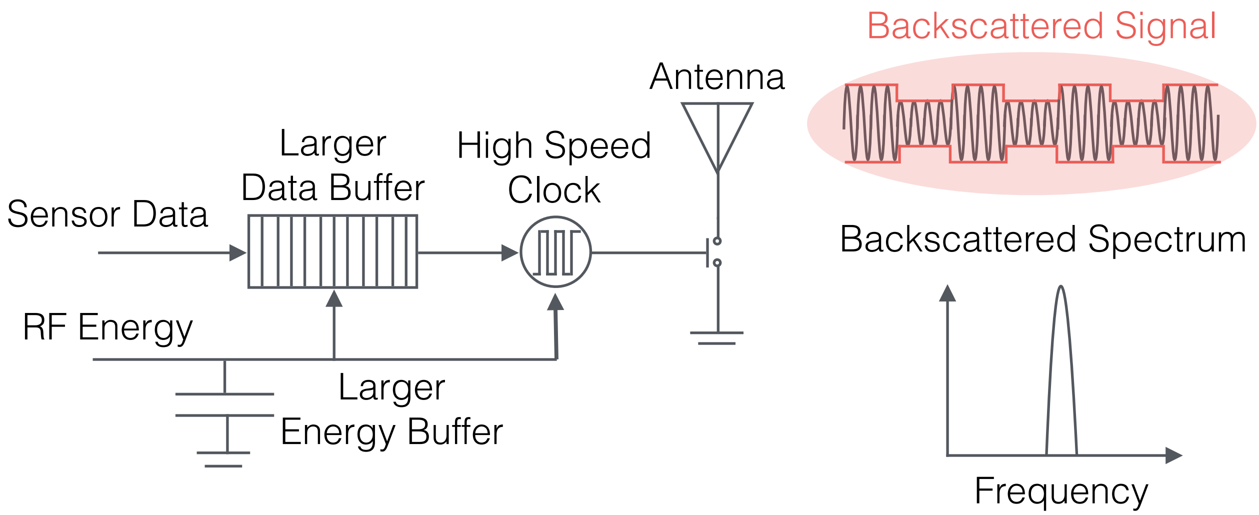
#2: Match bitrate to available spectrum



High power consumption

All these changes will result in higher power consumption.

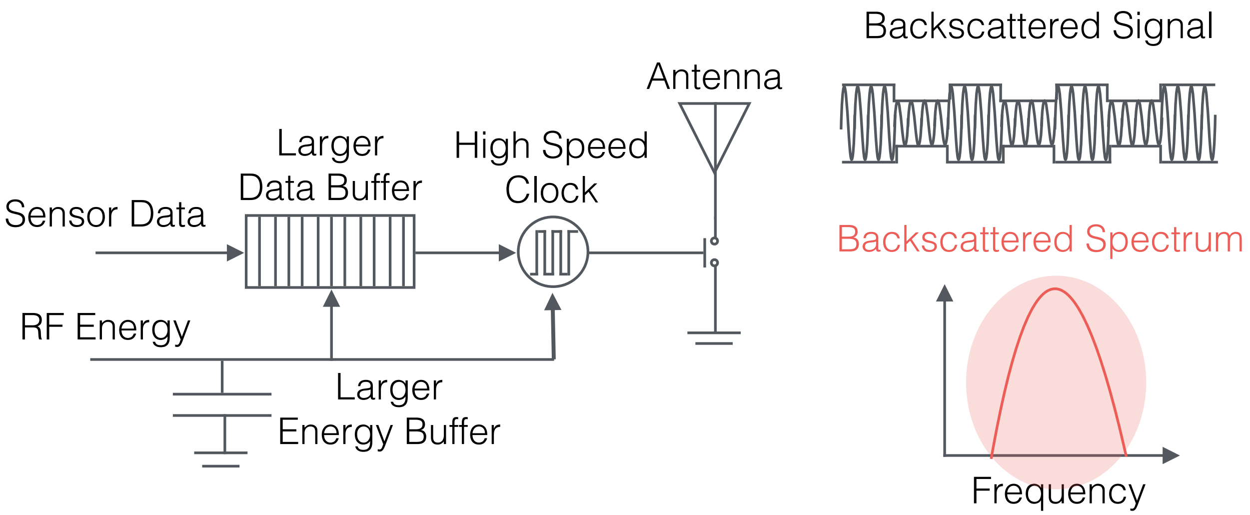
#2: Match bitrate to available spectrum



High power consumption

The good thing is that, since the backscatter transmit faster,

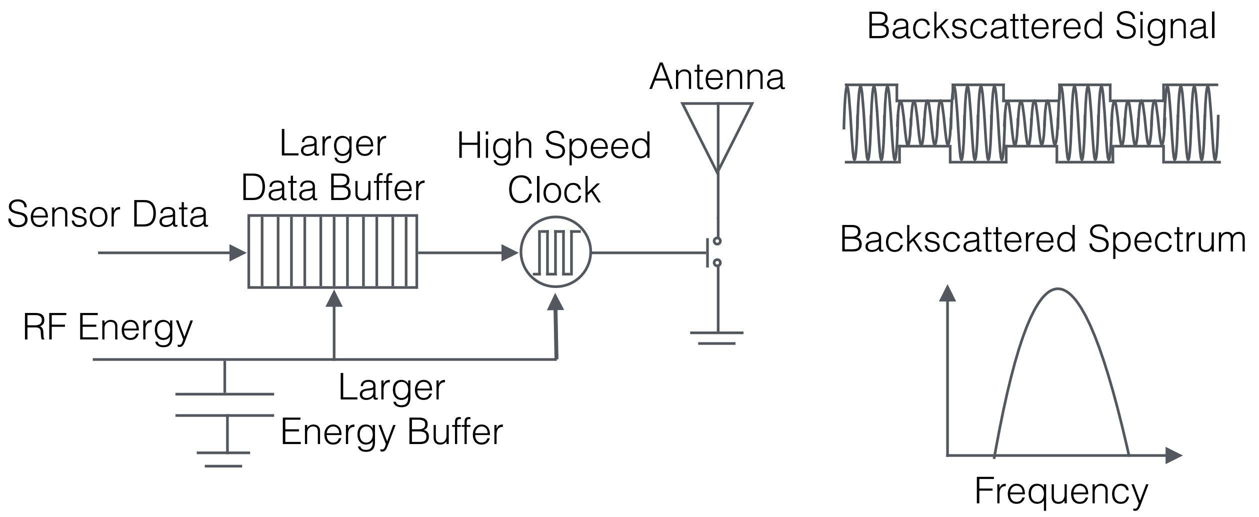
#2: Match bitrate to available spectrum



High power consumption

It can occupy larger spectrum

#2: Match bitrate to available spectrum

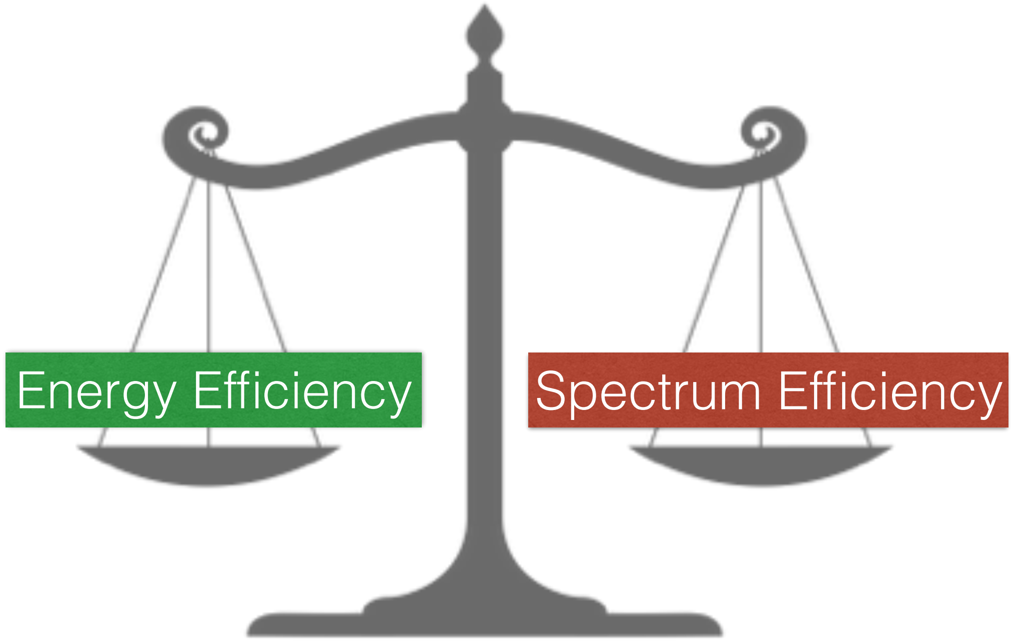


High power consumption

Better spectrum utilization

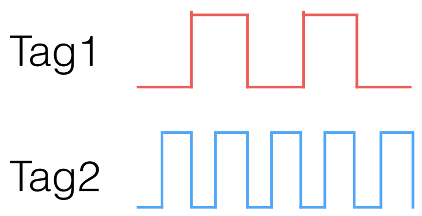
Leading to better spectrum utilization.

Energy v.s. Spectrum Tradeoff



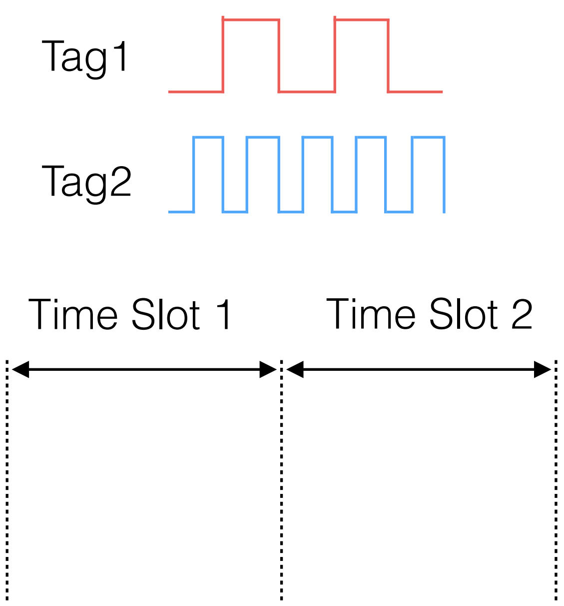
Given we have the tradeoff, let’s see how it affects the design of multiple access protocols.

TDMA - Sacrifice Spectrum Efficiency

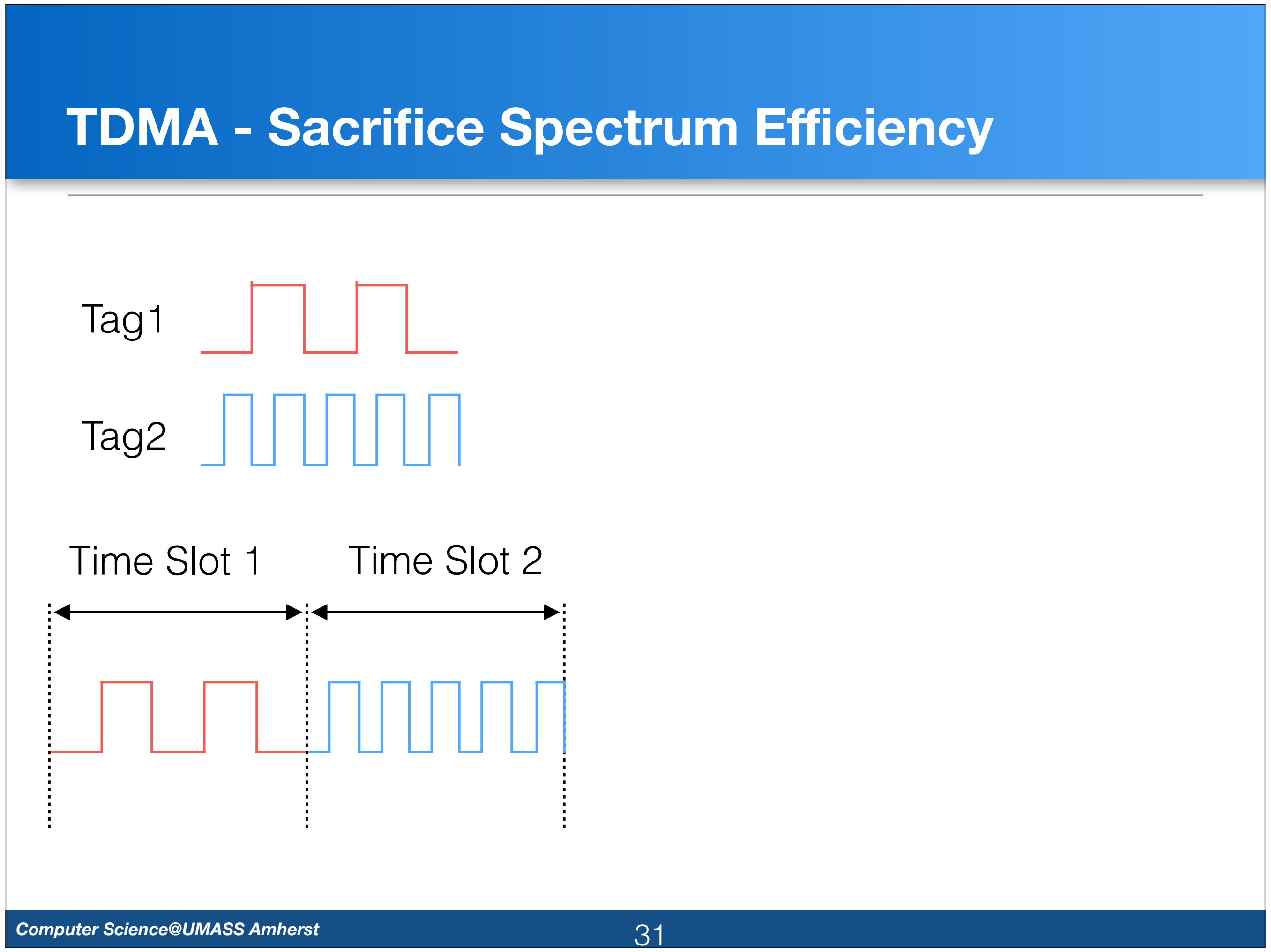


Let’s start with TDMA.

TDMA - Sacrifice Spectrum Efficiency



TDMA dives time into different slots



TDMA - Sacrifice Spectrum Efficiency

Tag1

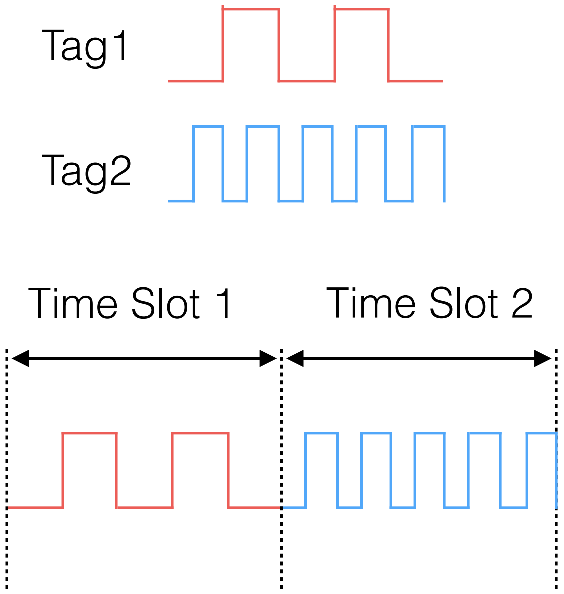
Tag2

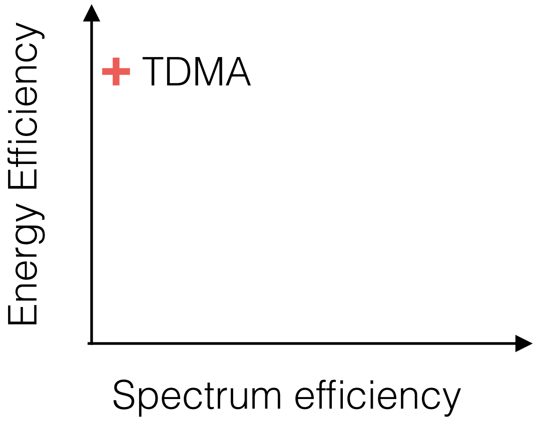
Time Slot 1

Time Slot 2

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.

TDMA - Sacrifice Spectrum Efficiency

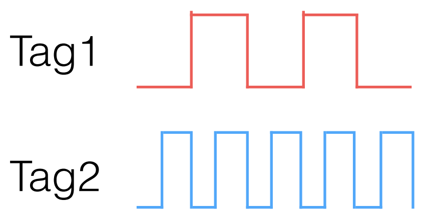




Low spectrum efficiency!

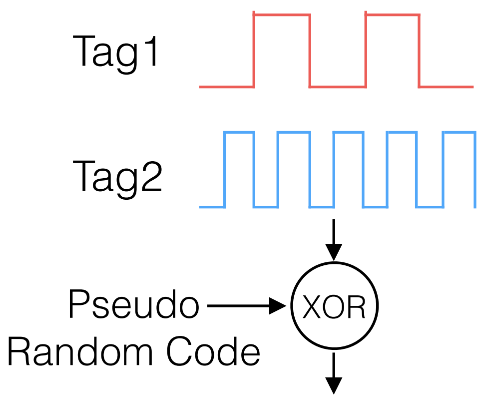
All these result in low spectrum efficiency. However, since each tag is transmit slowly, its energy efficiency is generally good.

CDMA - Sacrifice Power Efficiency



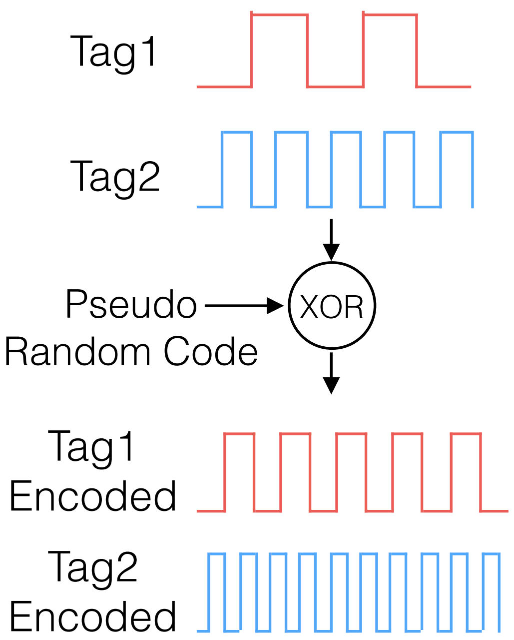
An alternative is CDMA.

CDMA - Sacrifice Power Efficiency



In CDMA, each bit will be expanded into multiple bits with a pseudo random code.

CDMA - Sacrifice Power Efficiency



Tag1

Tag2

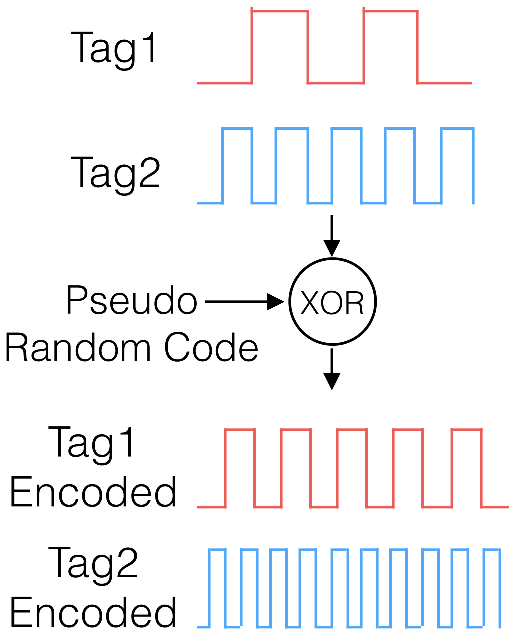
Pseudo Random Code

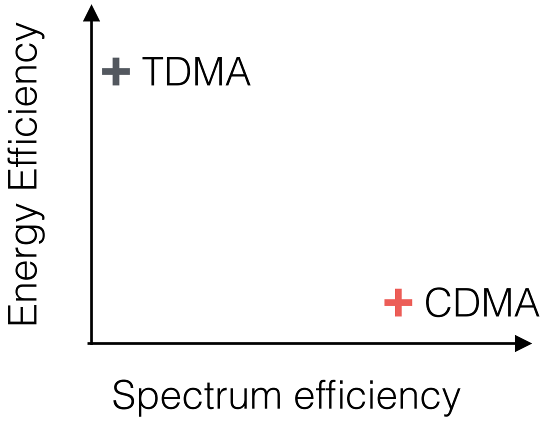
XOR

Tag1 Encoded

Tag2 Encoded

CDMA - Sacrifice Power Efficiency

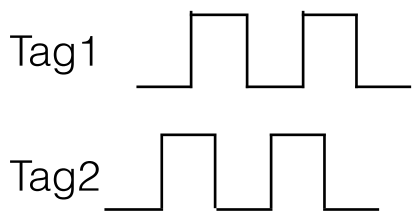




Low energy efficiency!

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.

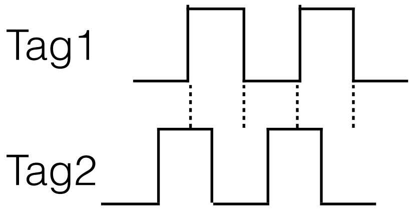
Buzz - Incurs Cost of Synchronization



Wang, Jue, et al. "Efficient and reliable low-power backscatter networks." Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication. ACM, 2012.

Another concurrent backscatter protocol is Buzz, which was presented a few years ago at SIGCOMM.

Buzz - Incurs Cost of Synchronization

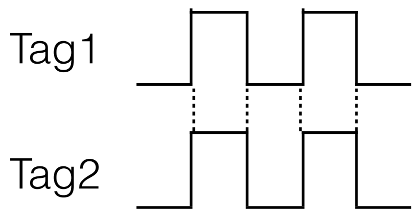


$$y=Ax$$

y: received signal
A: transmitted bits
x: channel matrix

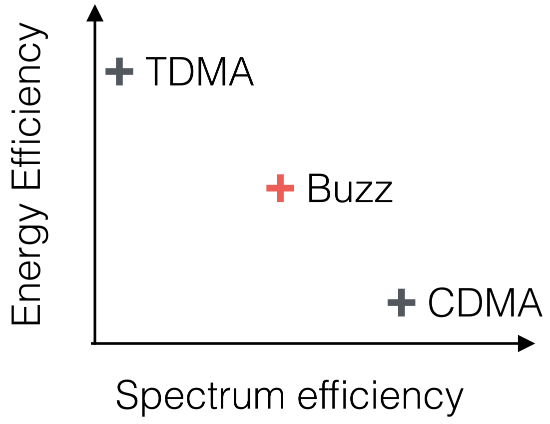
Buzz requires all the tags to toggle their transistors at the same time.

Buzz - Incurs Cost of Synchronization



y=Ax

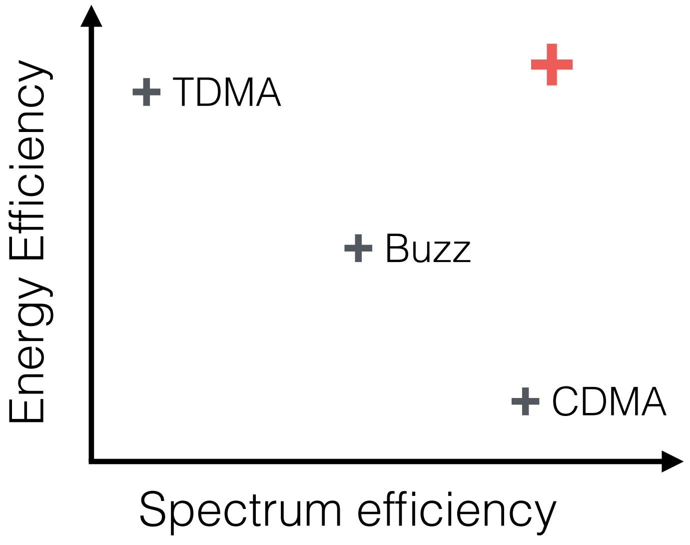
y: received signal
A: transmitted bits
x: channel matrix



At the cost of tight synchronization across tags

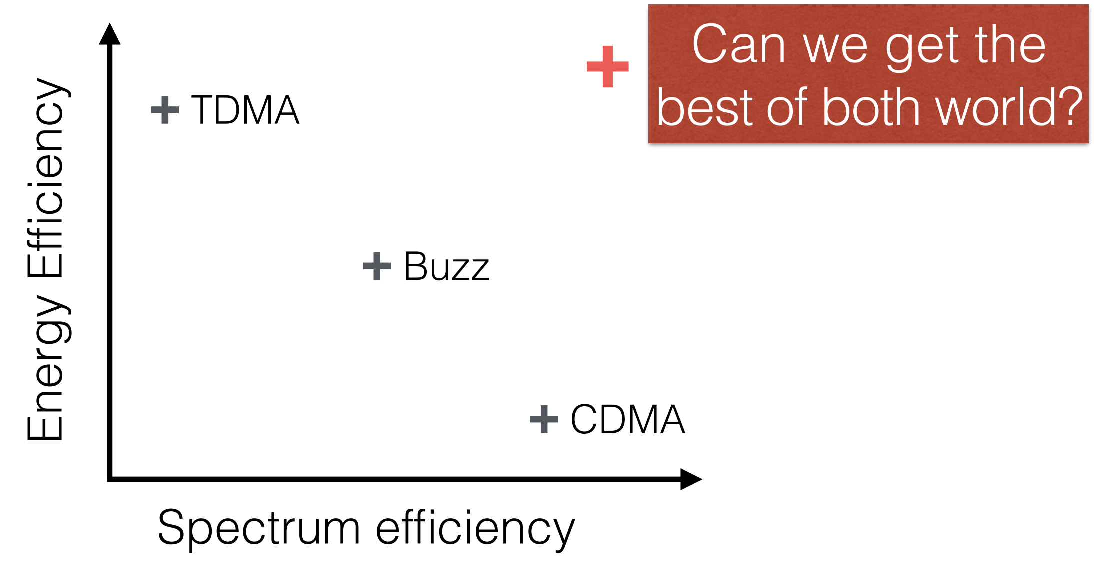
Buzz has better spectrum efficiency than TDMA but incurs the control overhead associated with synchronizing tags.

Higher bandwidth and energy efficiency?



So, there is a subtle tradeoff between power and bandwidth.

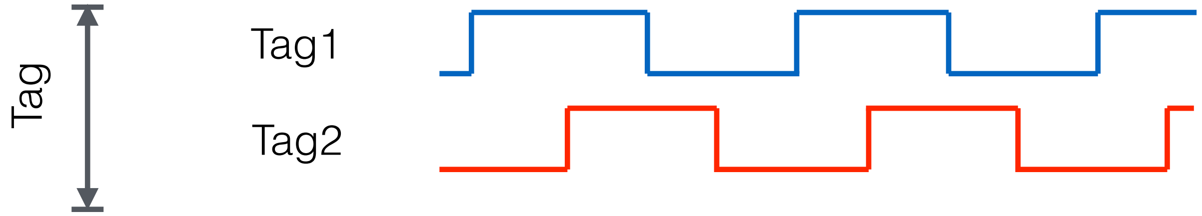
Higher bandwidth and energy efficiency?



Can we get the best of both world?

The question is that, can we get both energy and spectrum efficiency?

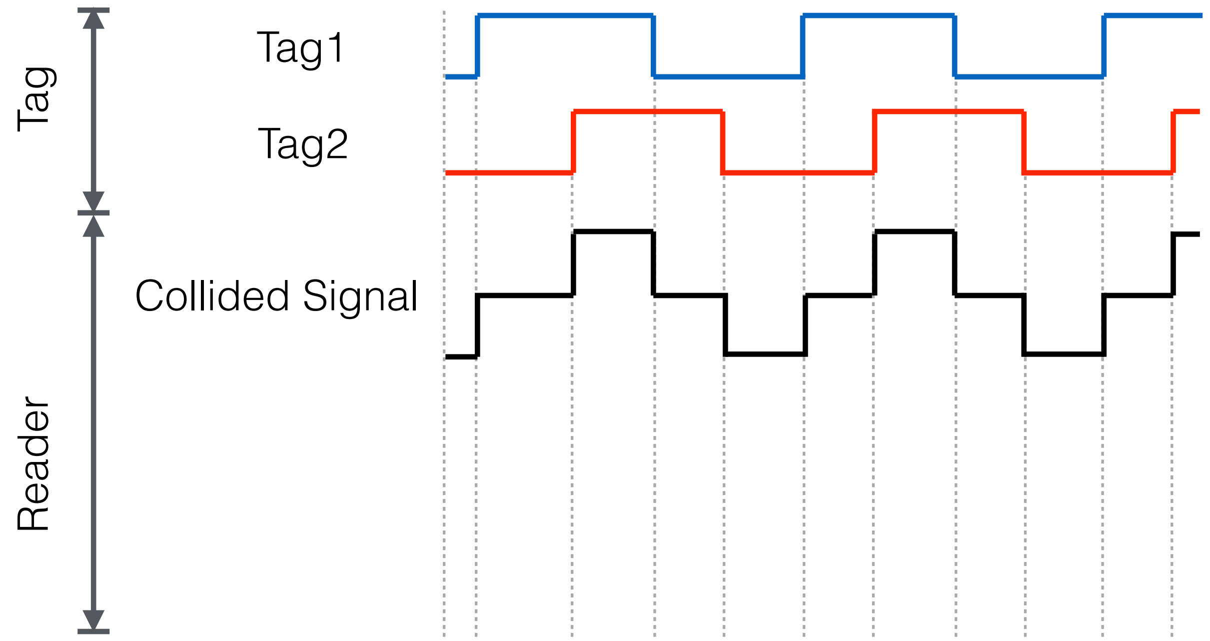
LF-Backscatter: Transmit at any time & any rate



Laissez Faire Backscatter:
A tag can transmit at any time and any bitrate.

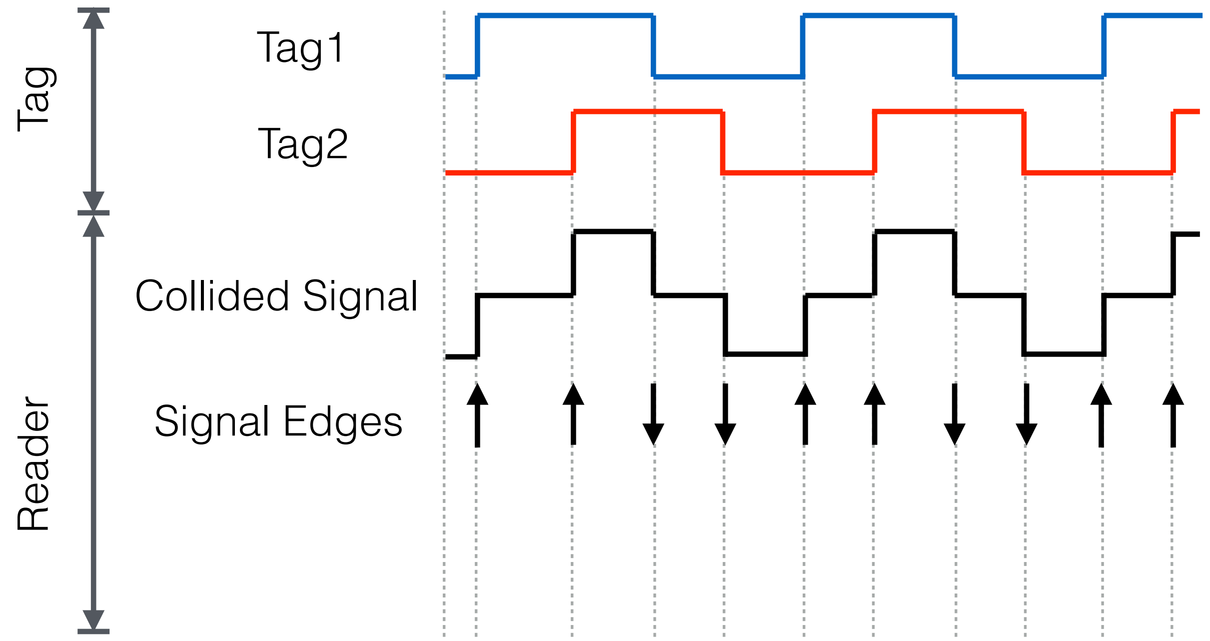
The answer is LF-Backscatter. A tag may start transmission at any time with any bitrate, therefore the term Laissez faire.

LF-Backscatter - Transmit Whenever It Wants To



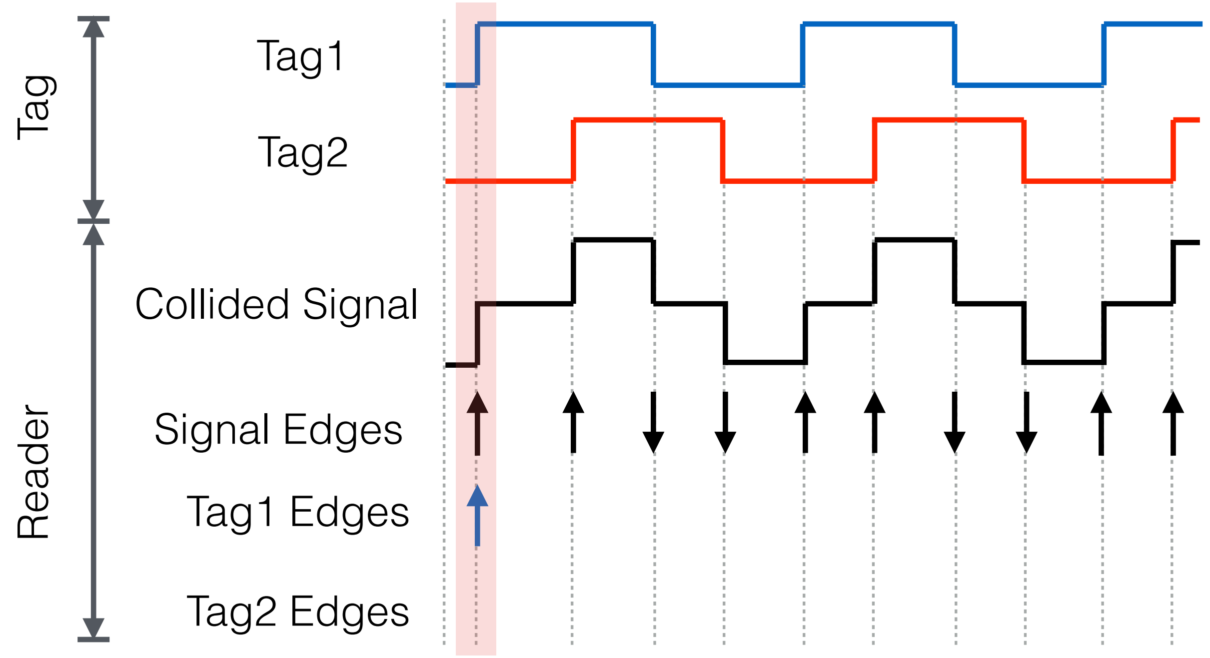
In LF-Backscatter, bits collision happens all the time.

LF-Backscatter - Transmit Whenever It Wants To



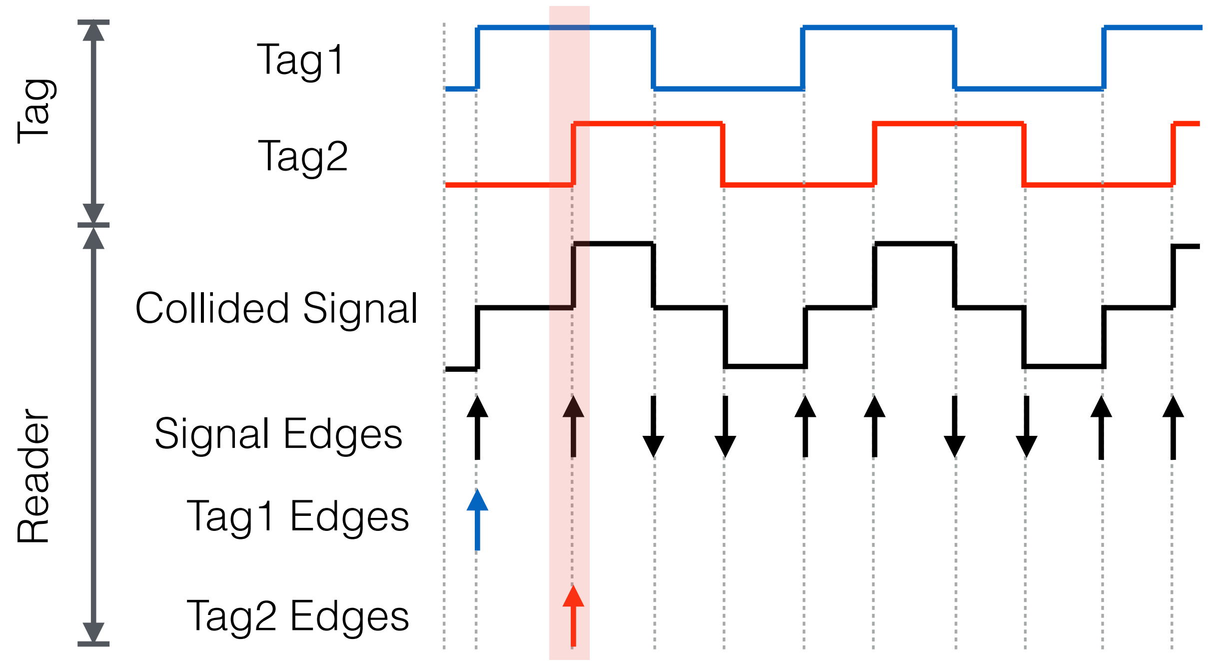
But how can we solve this problem? Take a closer look we can find that the edges in the collided signal still contains information from each tag.

LF-Backscatter - Transmit Whenever It Wants To



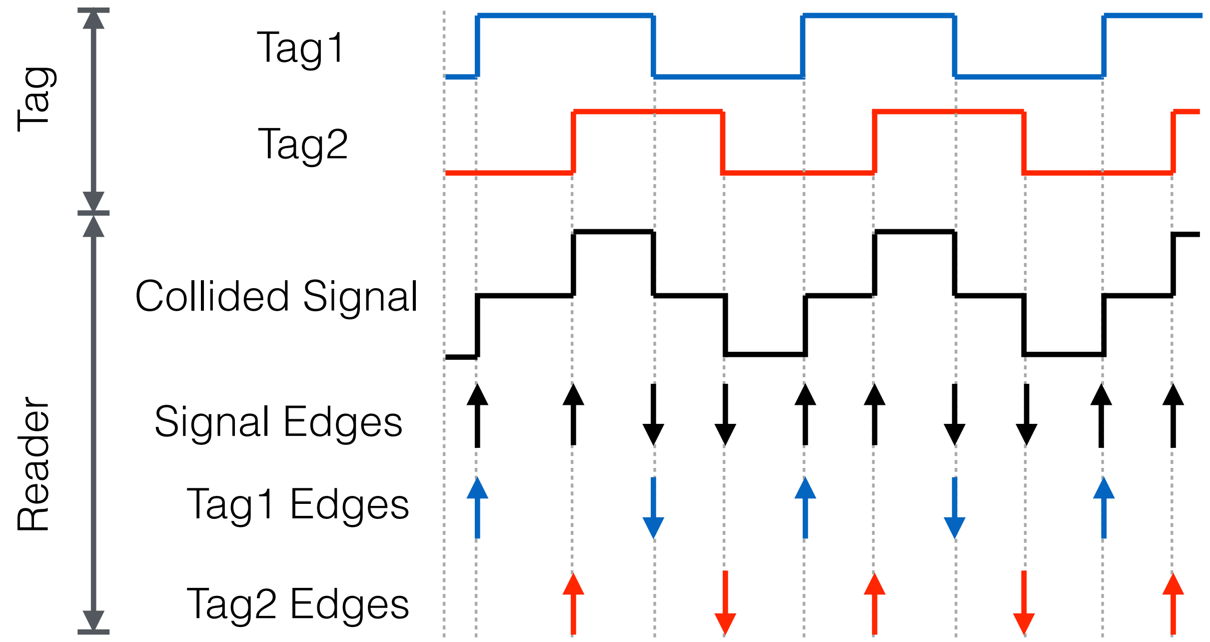
We can assign the edges to each of the tag

LF-Backscatter - Transmit Whenever It Wants To

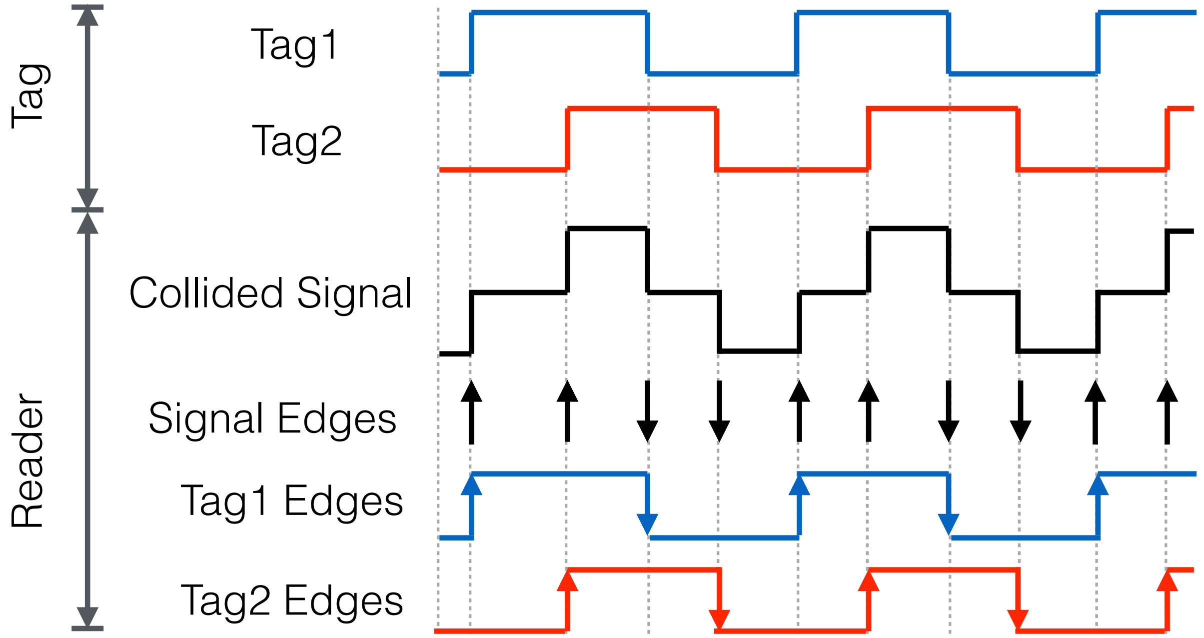


and do this repeatedly.

LF-Backscatter - Transmit Whenever It Wants To

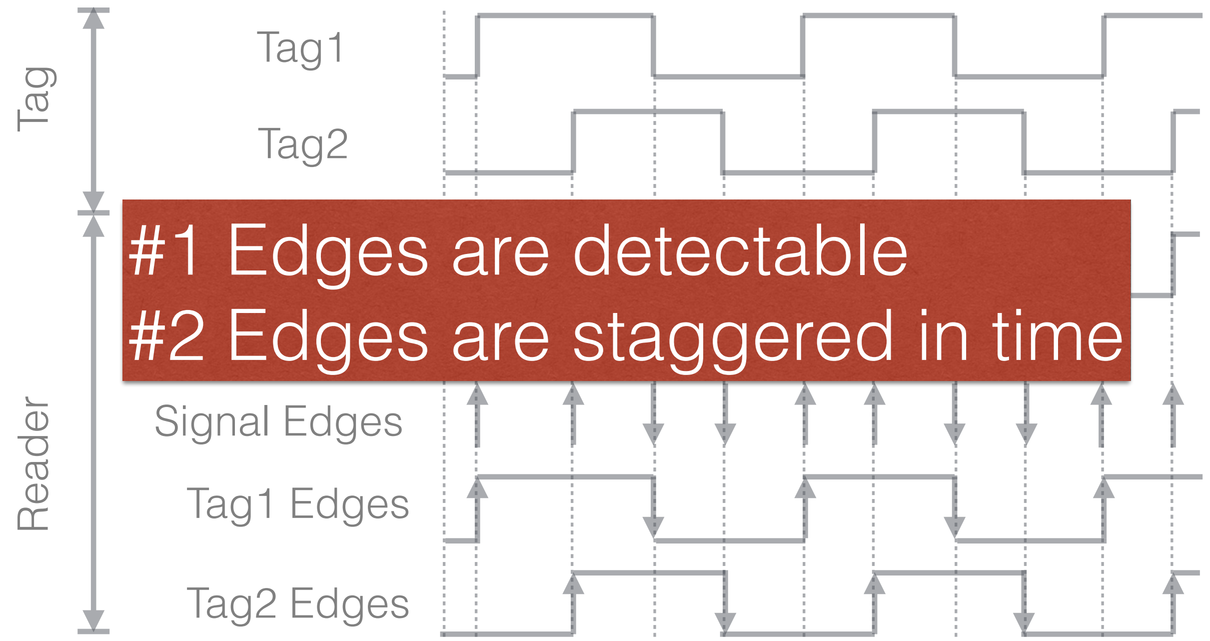


LF-Backscatter - Transmit Whenever It Wants To



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.

Two assumptions



#1 Edges are detectable
#2 Edges are staggered in time

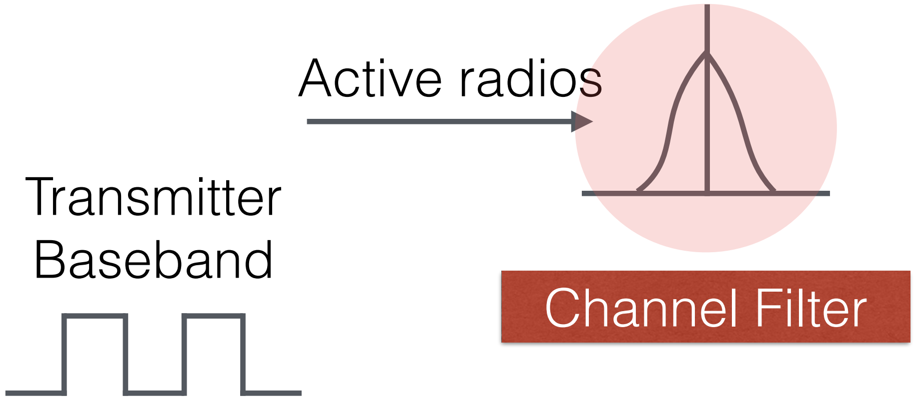
Here we made two assumption: the first one is, edges are detectable. The second one is, edges are staggered in time.

#1: Why are we able to detect signal edges?

Transmitter
Baseband

Let me explain them one by one. So why are we able to detect signal edges?

#1: Why are we able to detect signal edges?

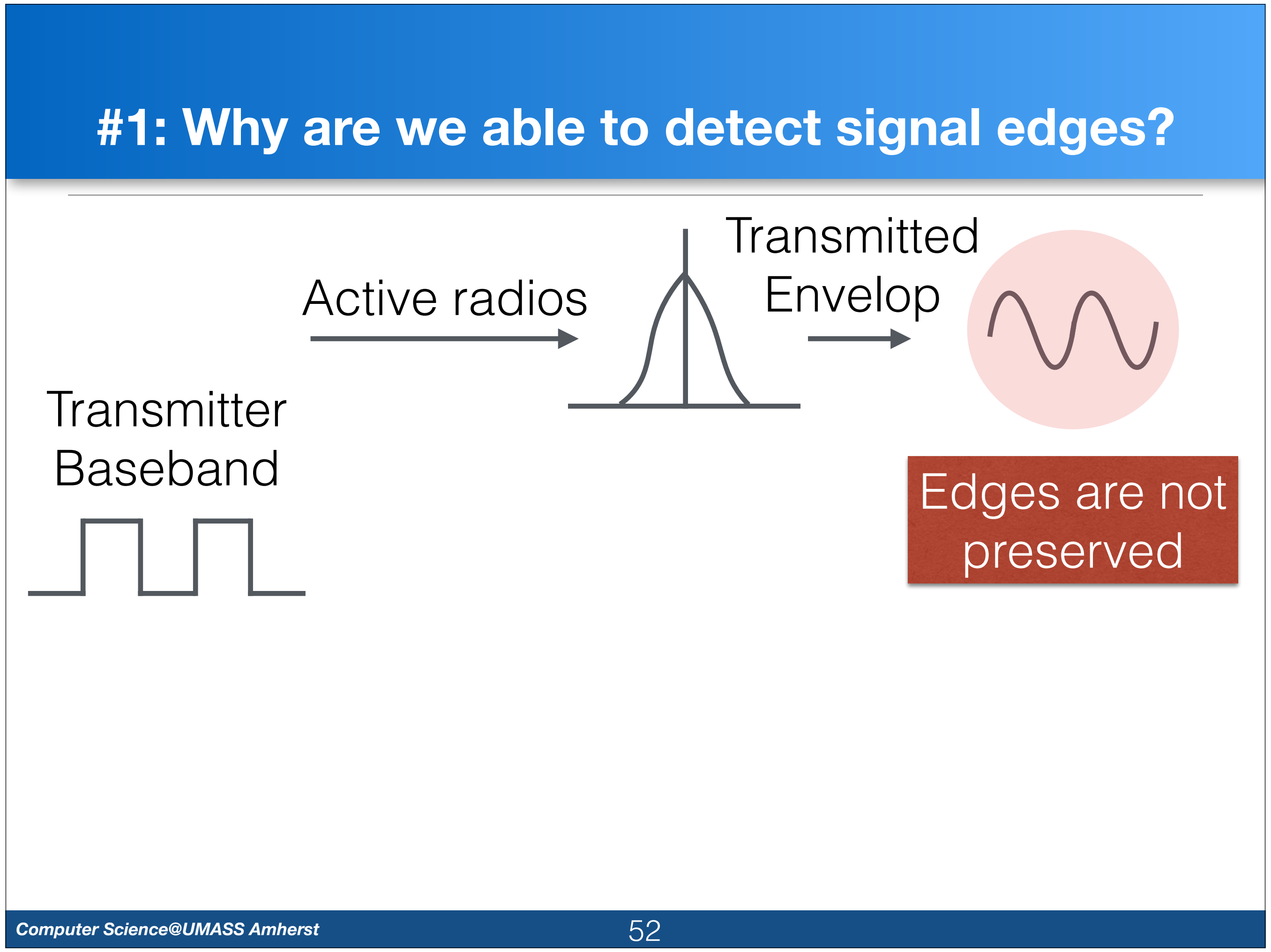


Active radios

Transmitter
Baseband

Channel Filter

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.



#1: Why are we able to detect signal edges?

Transmitter Baseband

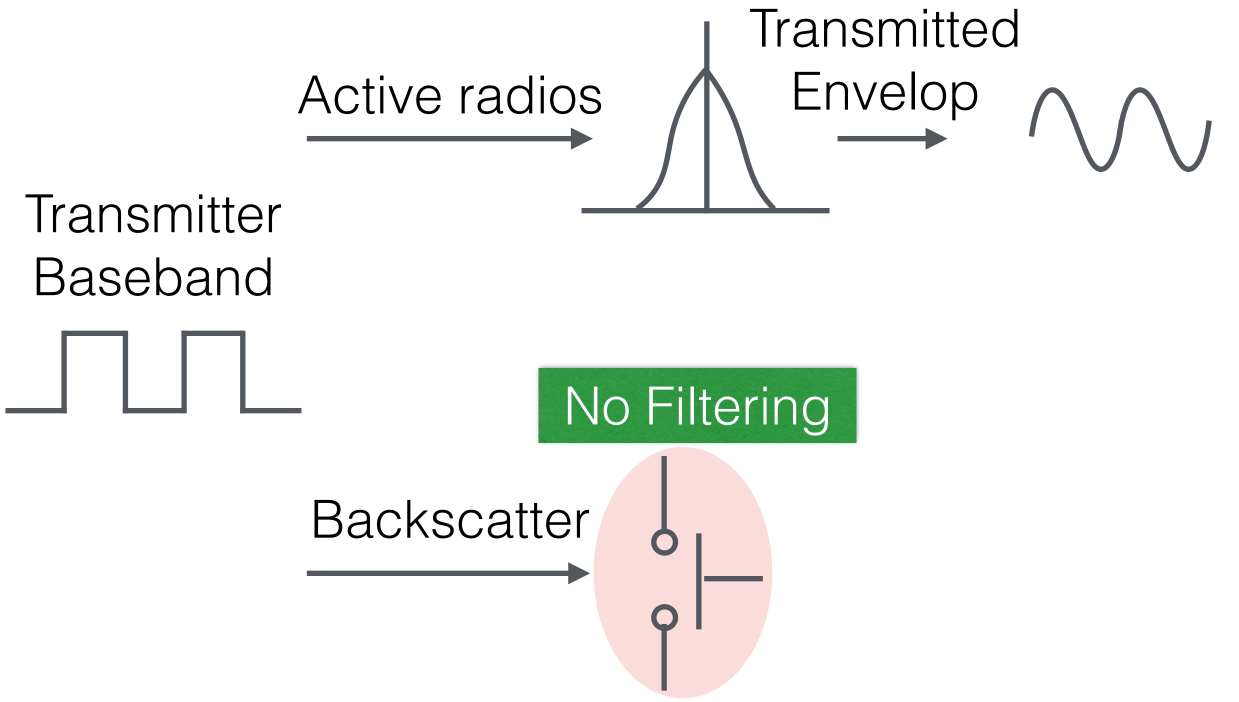
Active radios

Transmitted Envelop

Edges are not preserved

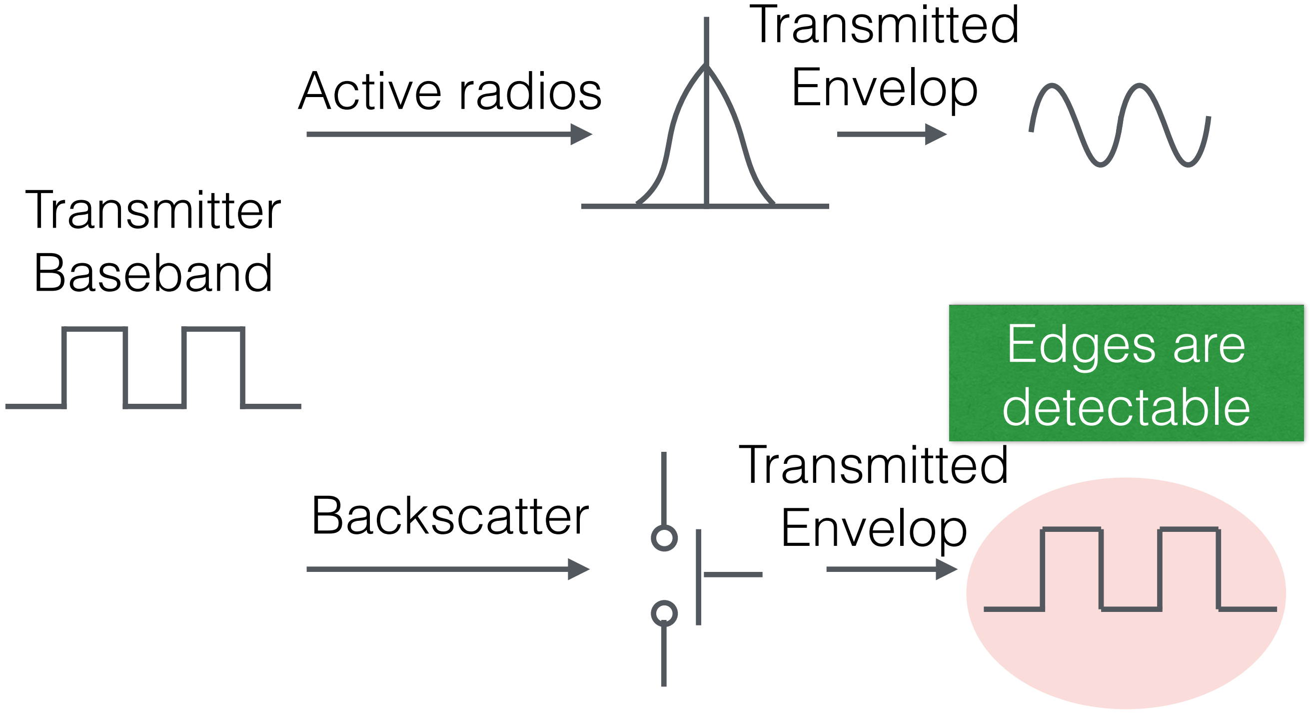
which reduces the sharpness of the signal edges, making edge detection impossible.

#1: Why are we able to detect signal edges?



However, there is no such filtering in backscatter.

#1: Why are we able to detect signal edges?



Transmitter Baseband

Active radios

Transmitted Envelop

Backscatter

Transmitted Envelop

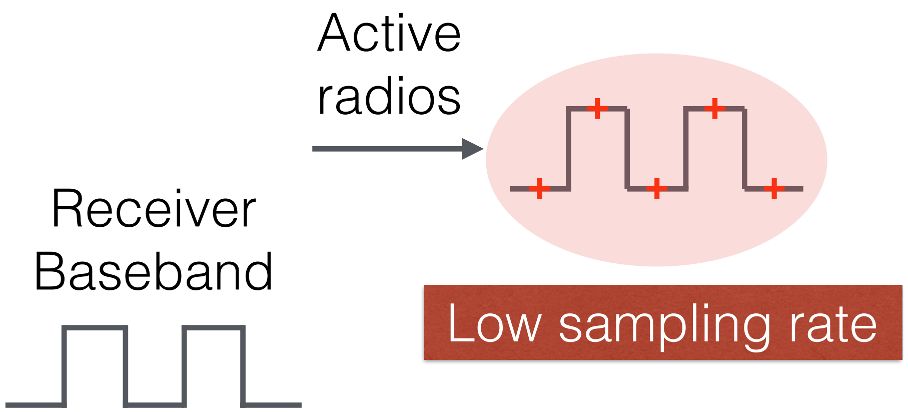
Edges are detectable

So the edges are clearly detectable.

#1: Why are we able to detect signal edges?

Let’s now look at the receiver side. The receiver also should have the capability to detect signal edges.

#1: Why are we able to detect signal edges?



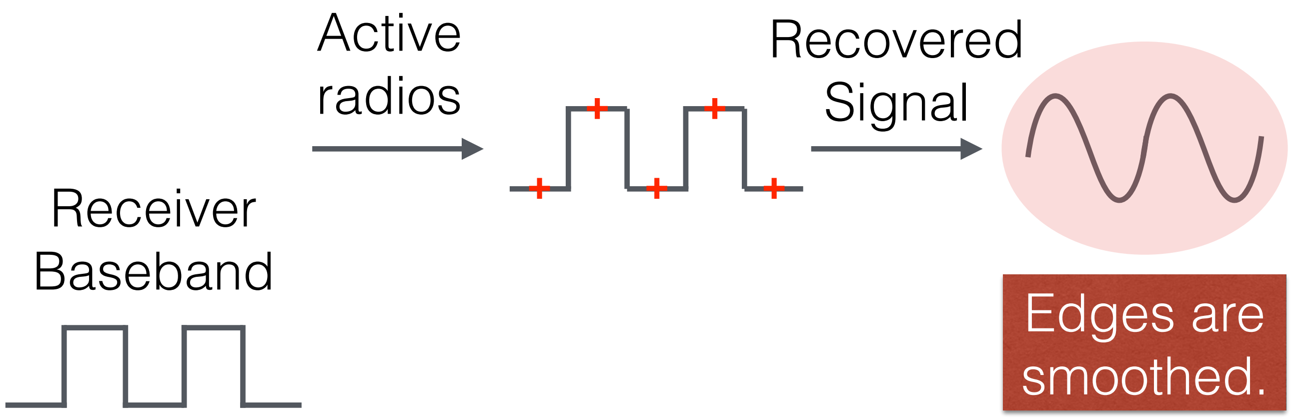
Active radios

Receiver Baseband

Low sampling rate

In an active radio, the receiver typically sampling at the rate which is usually similar as the transmission bit rate.

#1: Why are we able to detect signal edges?



Receiver Baseband

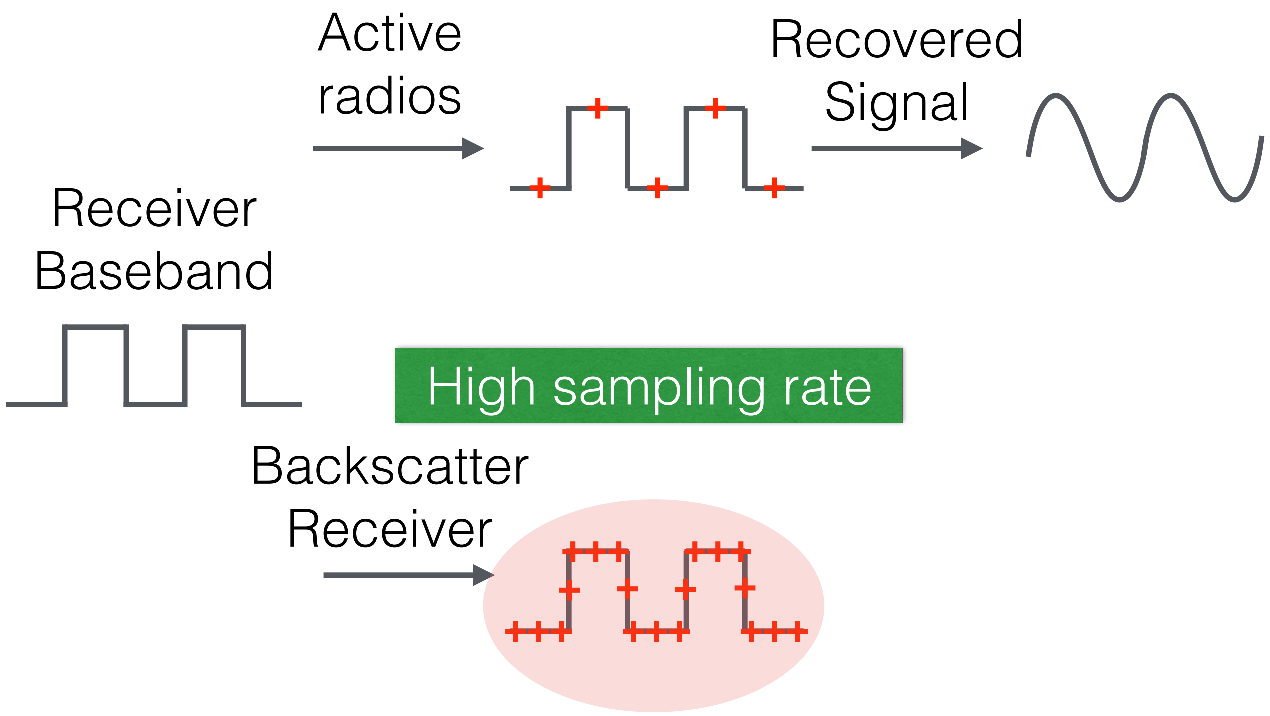
Active radios

Recovered Signal

Edges are smoothed.

The low sampling rate leads to smoothed edges in recovered signal.

#1: Why are we able to detect signal edges?



Receiver Baseband

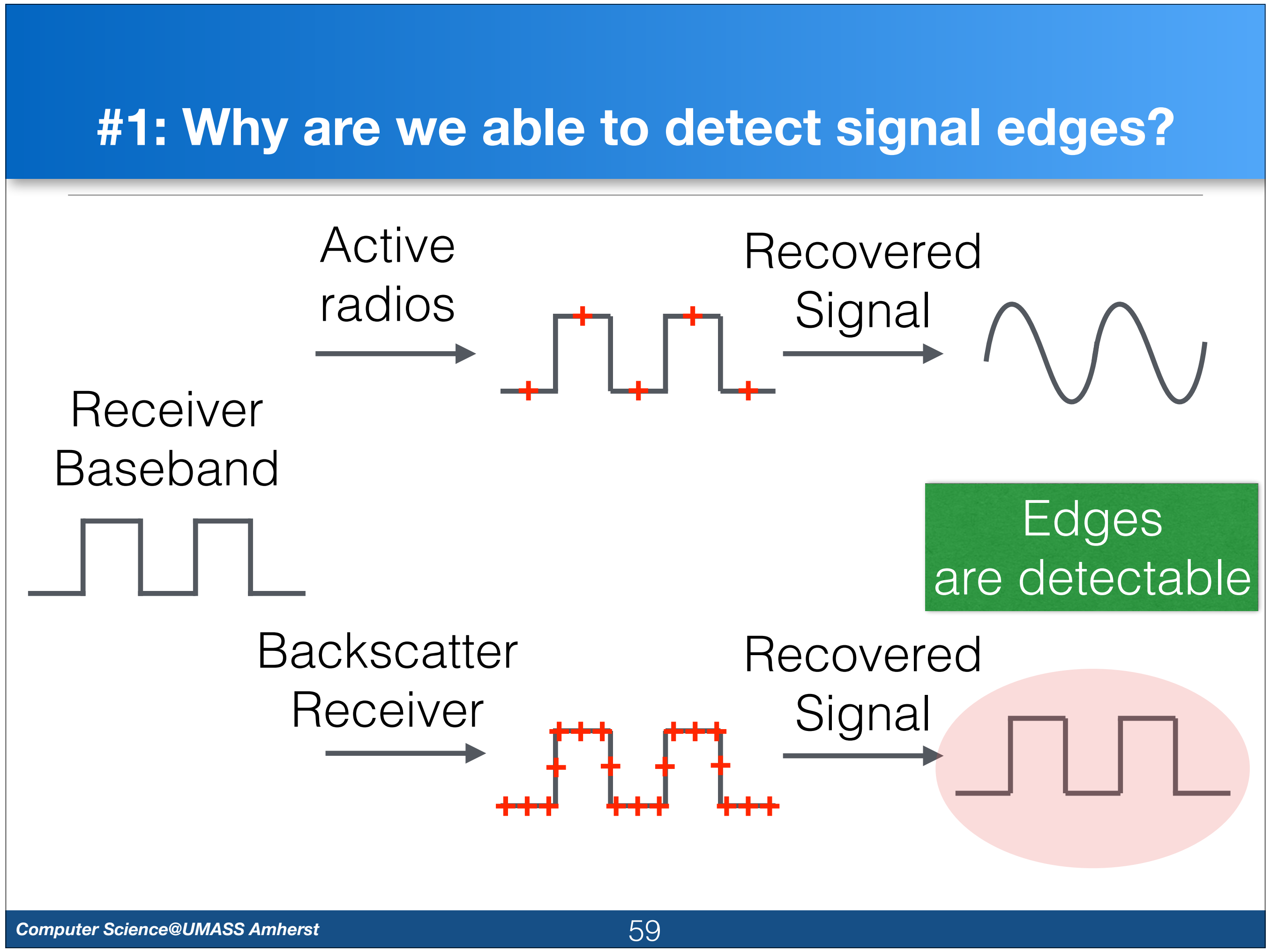
Active radios

Recovered Signal

High sampling rate

Backscatter Receiver

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.



#1: Why are we able to detect signal edges?

Receiver Baseband

Active radios

Recovered Signal

Edges are detectable

Backscatter Receiver

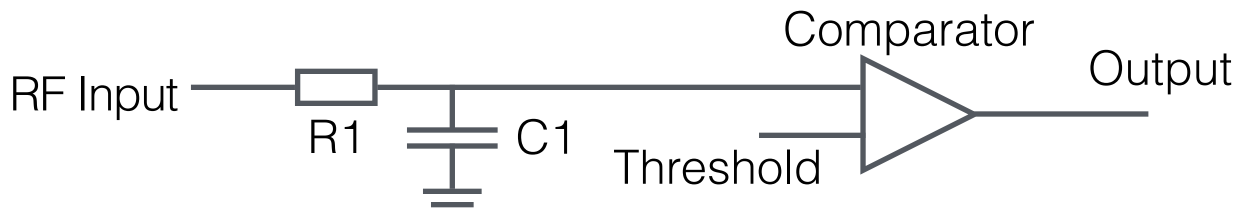
Recovered Signal

In this case, there are many sample before, on top of and after the edge. So the edges can be detected.

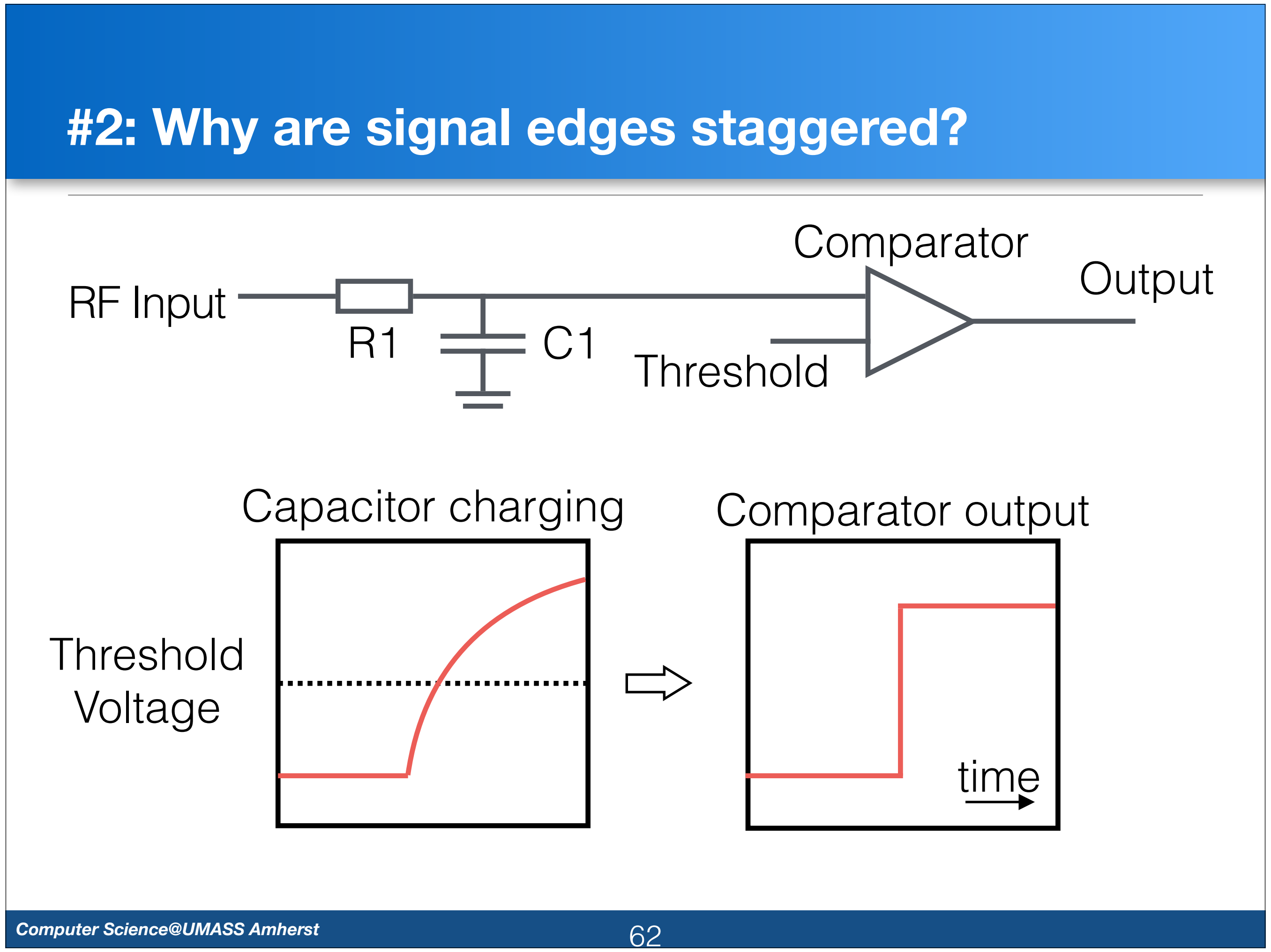
#2: Why are signal edges staggered?

Our second assumption is that, edges are staggered.

#2: Why are signal edges staggered?



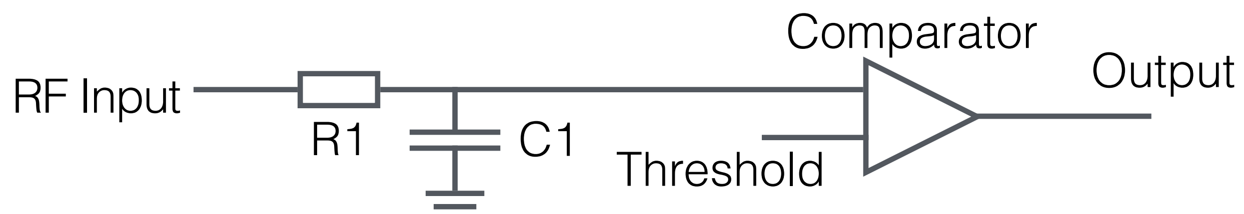
Why this is true? This actually occurs quite naturally because of the way a backscatter receiver works. A backscatter receiver is a simple envelope detector

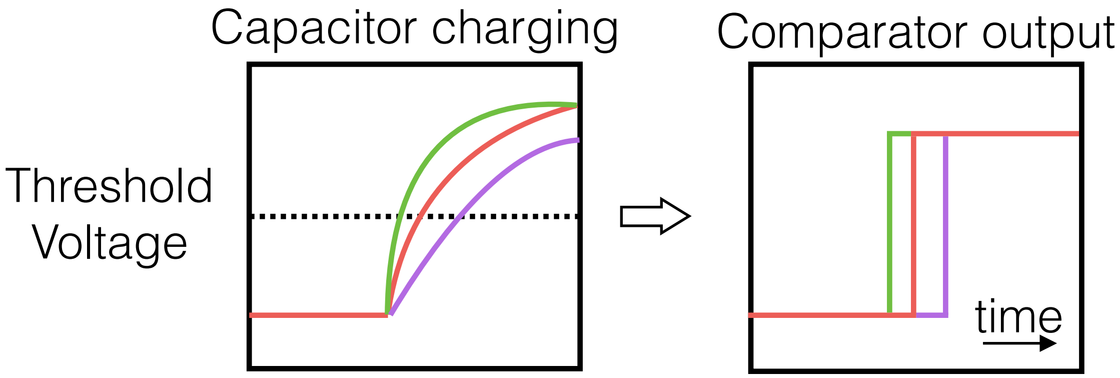


#2: Why are signal edges staggered?

It detected the presence of the reader when the capacitor charges to a certain threshold, after than the comparator generate an output and the tags start to transmit.

#2: Why are signal edges staggered?



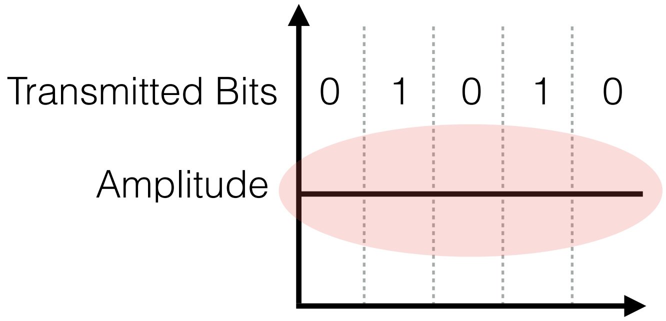


However, tags at different locations can have different signal strength, affecting the capacitor's charging speed. In addition to that, there are about 20% variation in terms of capacitance when manufacturing the capacitors. All these factors will result in different charging curves which result in different start time. As a result, the edges are naturally staggered.

Robust edge detection using IQ vector

We’ve seen is that in theory, we can detect signal edges, but is this method sufficiently robust in practice?

Robust edge detection using IQ vector

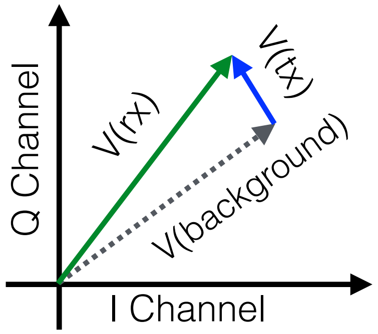


Amplitude may not change during transmission.

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.

Robust edge detection using IQ vector

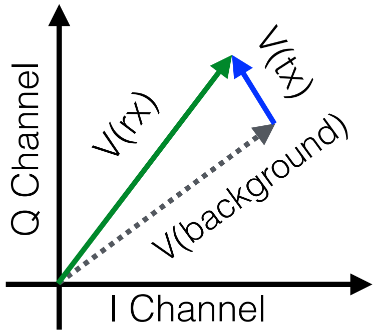
When Tag transmitting 1:



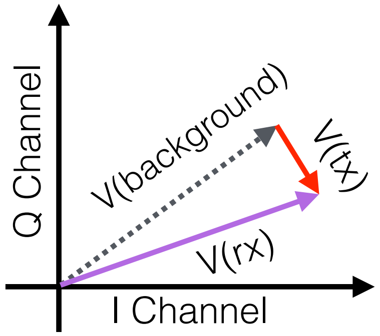
Why is this the case? Let’s look at the IQ signal plot when the tag is transmitting 1s. Firstly we will have a strong background signal due to self-interference and environment reflection. Also, we have the signal transmitted by the tag, as shown in blue. The received signal is a combination of both.

Robust edge detection using IQ vector

When Tag
transmitting 1:

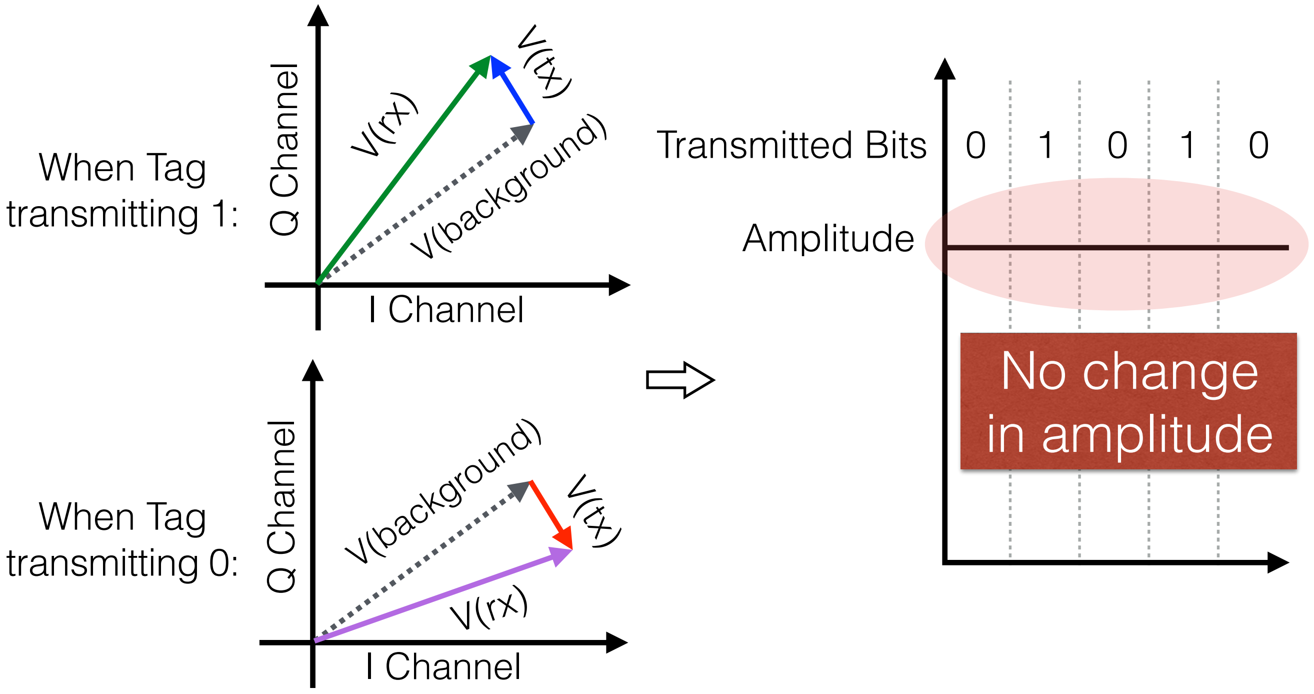


When Tag
transmitting 0:



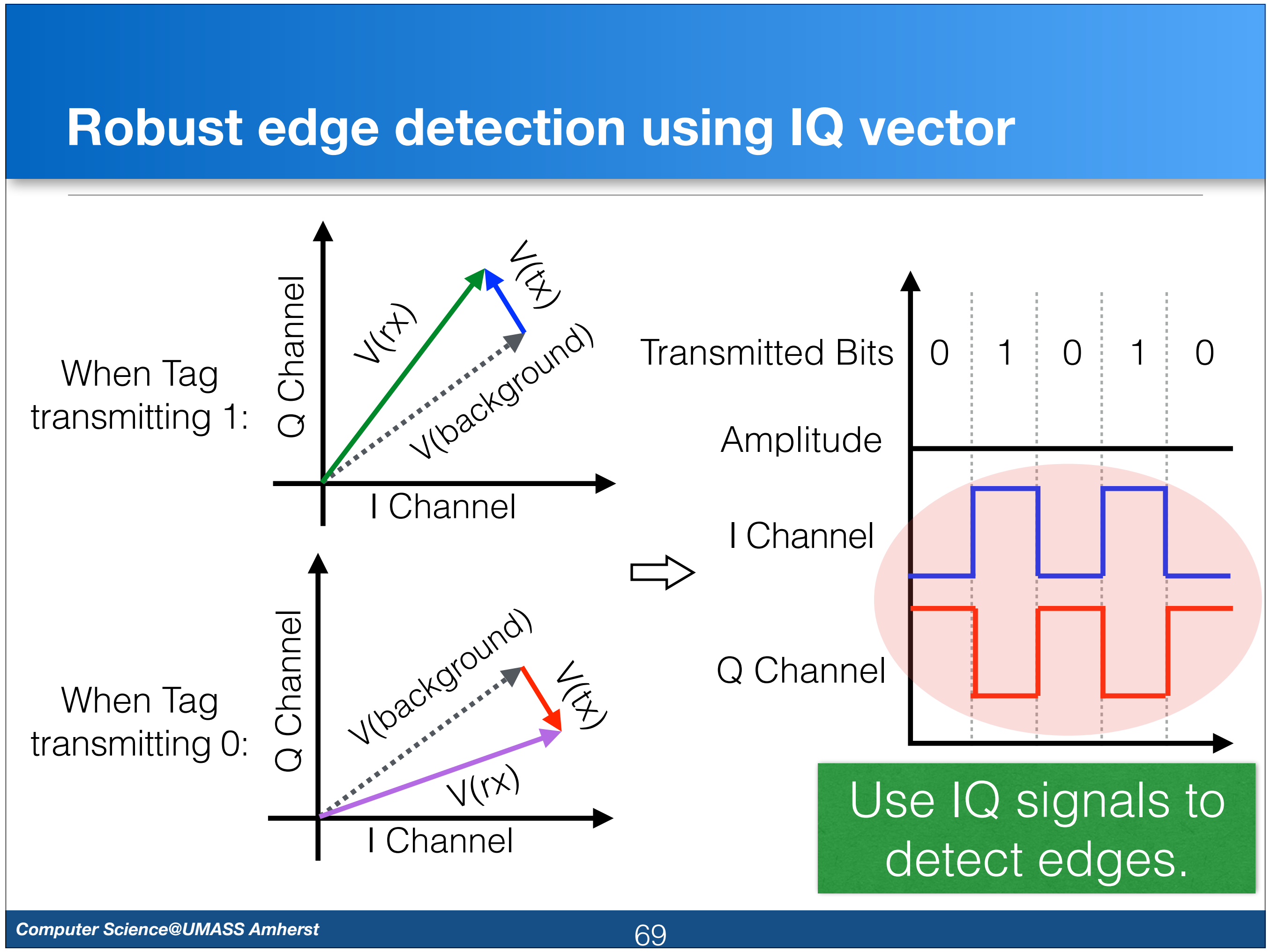
Similarly, we can have the IQ plot when the tag is transmitting 0s. But what will happen if the signal transmitted by the tag is orthogonal to the backsground?

Robust edge detection using IQ vector



No change in amplitude

The result will be that there is no change in amplitude regardless of the bits transmitted.



Robust edge detection using IQ vector

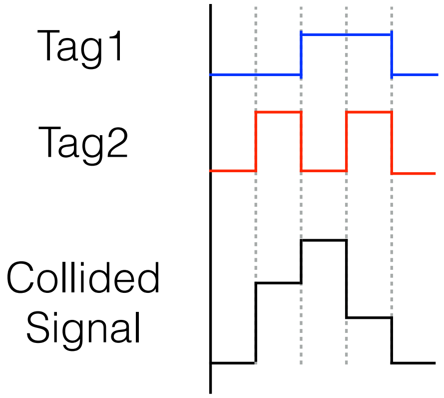
Use IQ signals to detect edges.

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?

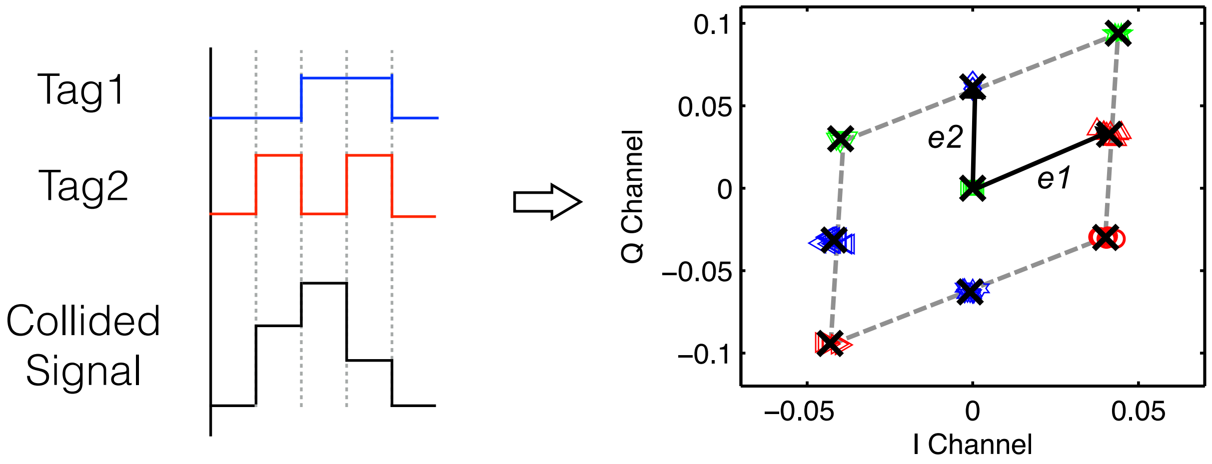
So far, I've assumed that that signal edges DO NOT collide with each other, but edges can collide especially when there is a large number of tags. One approach is retransmissions, but in many cases, we find that this may not be necessary.

How to deal with edge collisions?



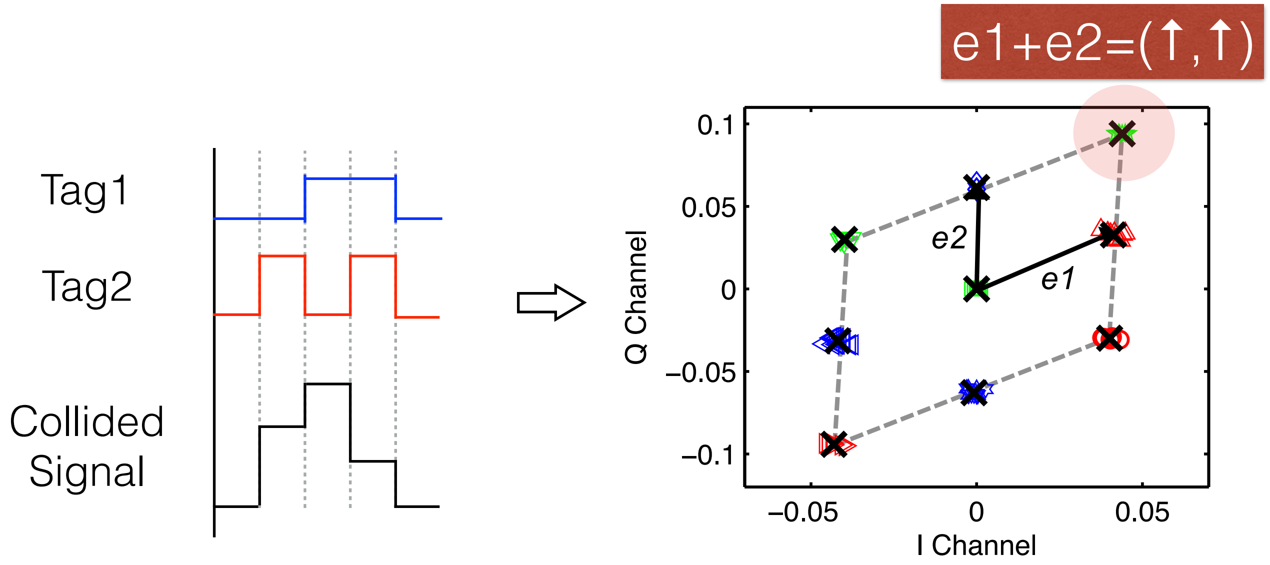
Let us look at this example where the two signal edges collides.

How to deal with edge collisions?



Assuming that two tags have different signal edges e1 and e2. 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.

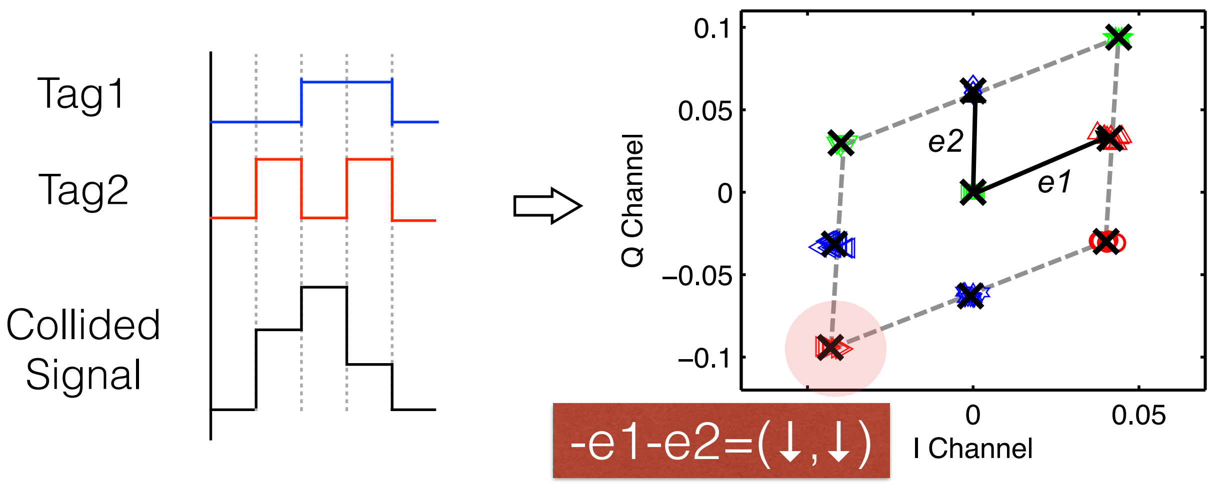
How to deal with edge collisions?



e1+e2=(↑,↑)

For example, the top cluster is contributed by two rising edges

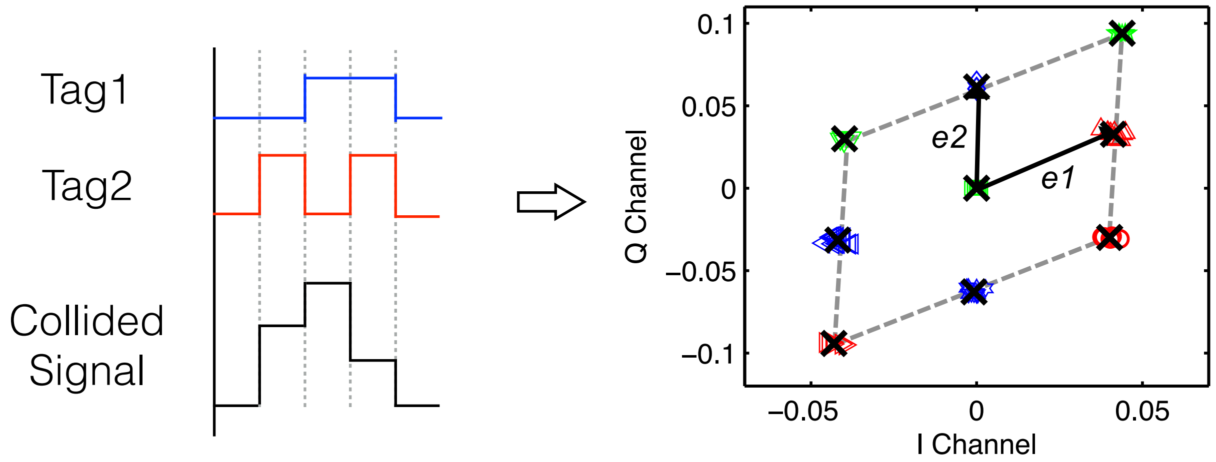
How to deal with edge collisions?



-e1-e2=(↓,↓)

and the bottom cluster is caused by falling edges.

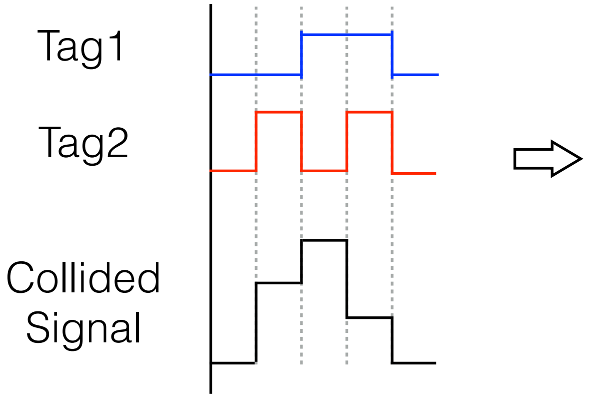
How to deal with edge collisions?

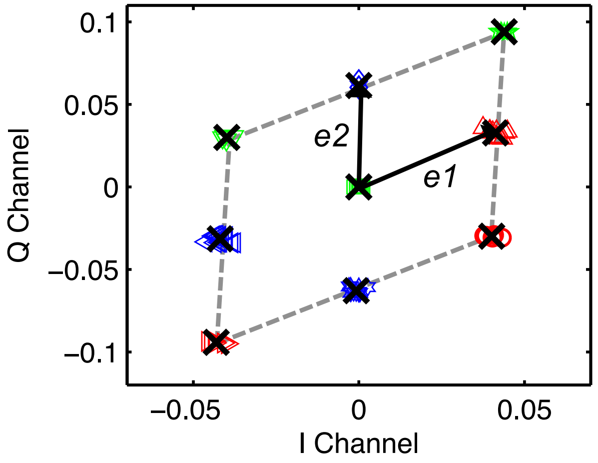


9 clusters

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.

How to deal with edge collisions?





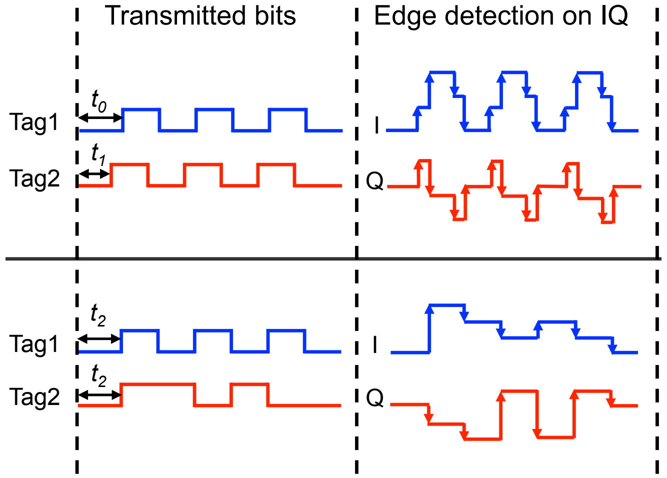
9 clusters

More clusters?

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.

LF-Backscatter: The Big Picture

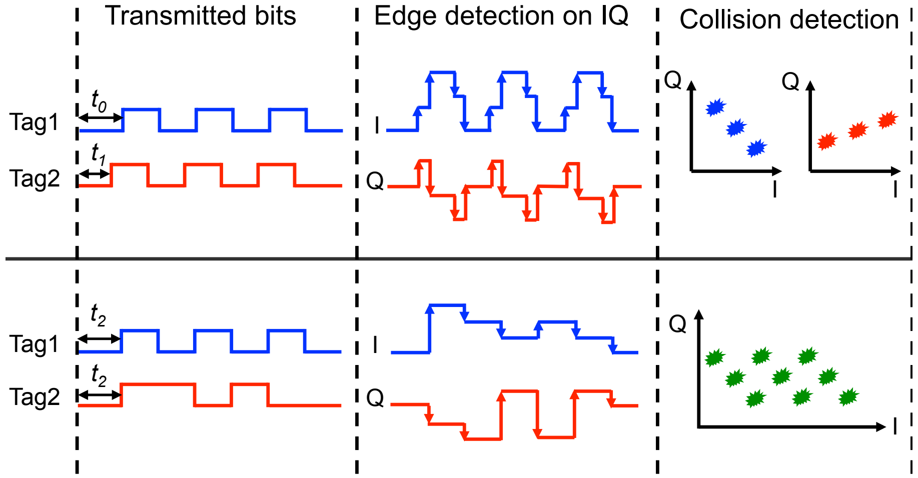
- IQ-Based edge detection
- Collision detection and recovery
- Decoding



Lets put everything together and provide a complete picture of LF-Backscatter. LF-Backscatter starts by detecting signal edges in IQ domain,

LF-Backscatter: The Big Picture

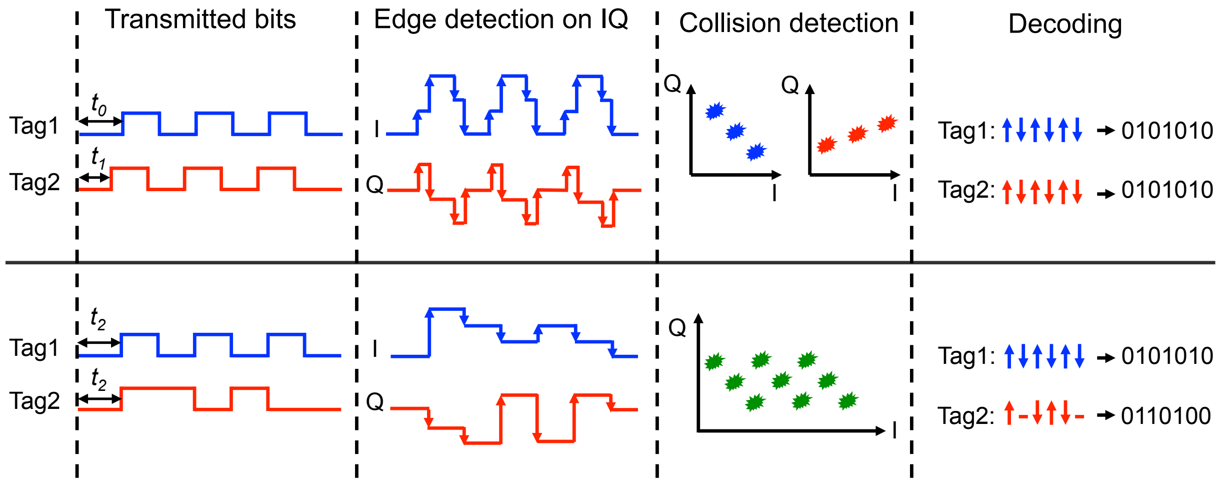
- IQ-Based edge detection
- Collision detection and recovery
- Decoding



then detect collisions and resolve collision with IQ cluster information.

LF-Backscatter: The Big Picture

- IQ-Based edge detection
- Collision detection and recovery
- Decoding

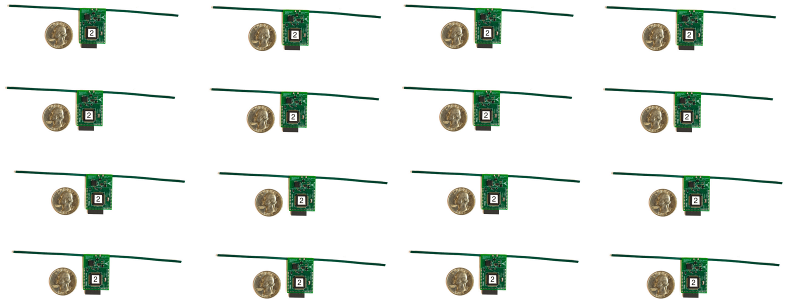


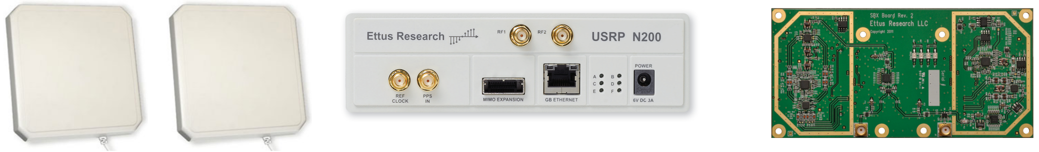
After that, LF Backscatter assign edges to each tags and decode the data.

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.

Evaluation

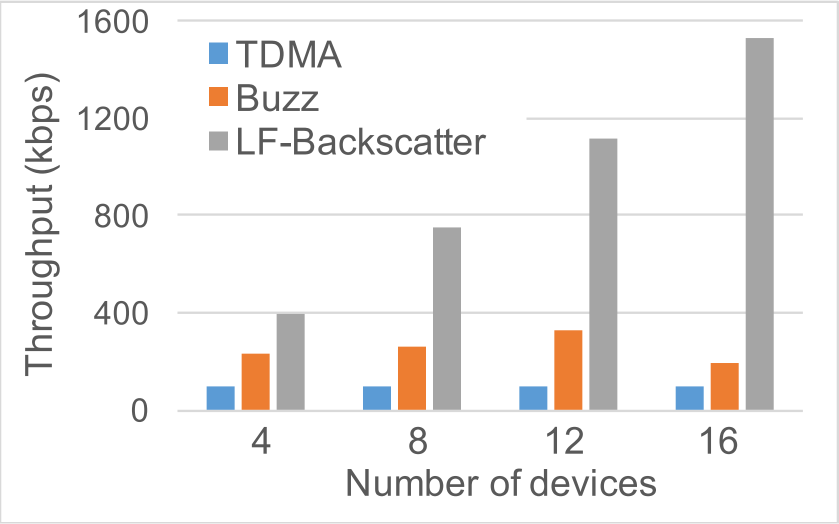
- 16 UMass Moo Tags
- USRP N210 + SBX Daughterboard





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.

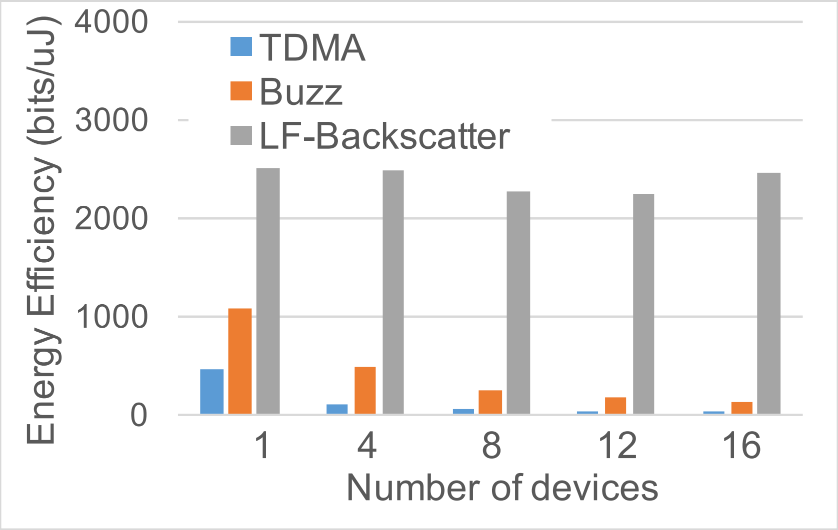
Throughput



15x better than TDMA and 7x better than Buzz.

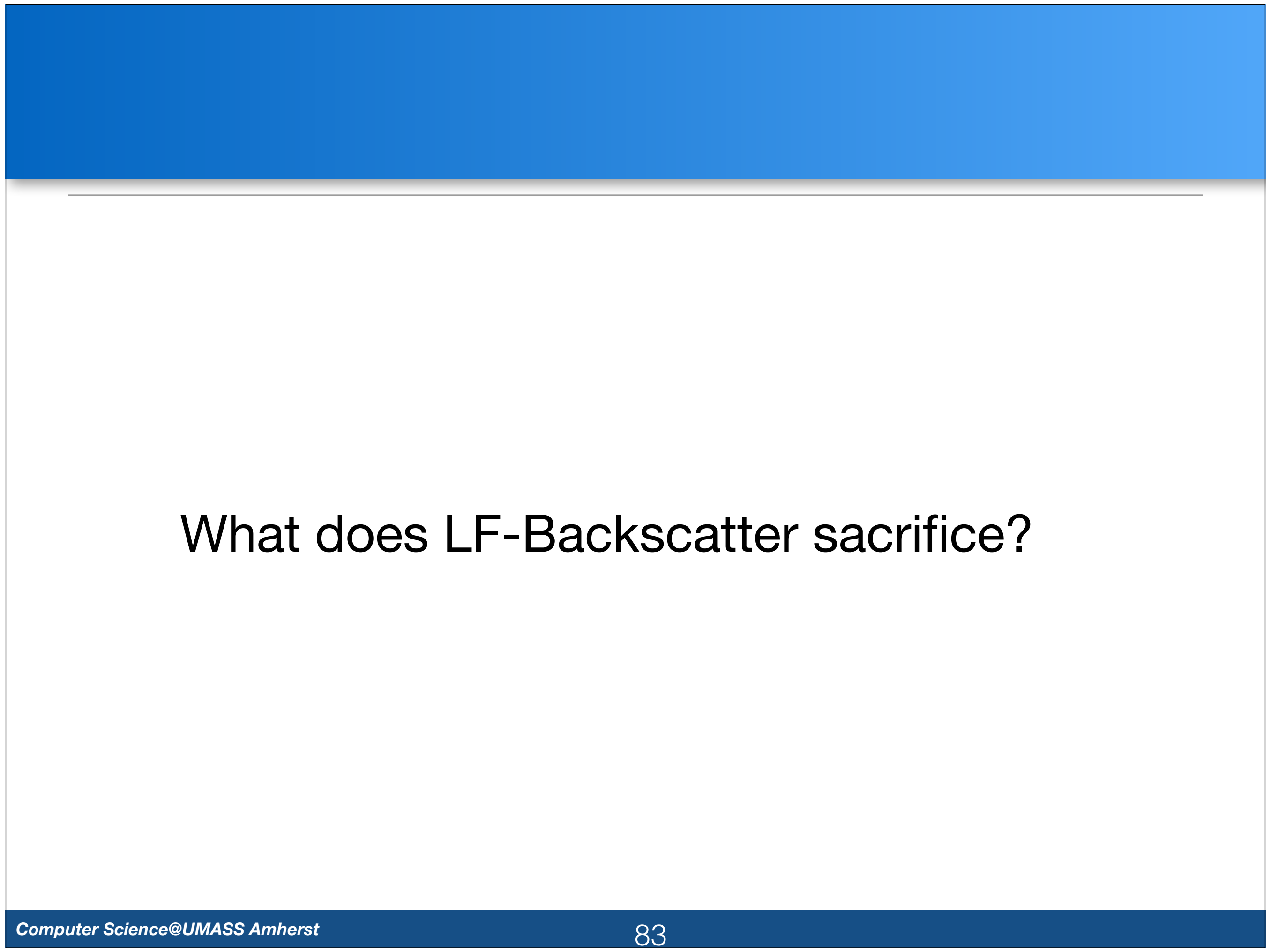
Our experiment shows that LF-Backscatter is able to achieve 15x throughput improvement over TDMA, and 7x over Buzz.

Energy-efficiency



Scalable. 20x more energy-efficient than TDMA/Buzz

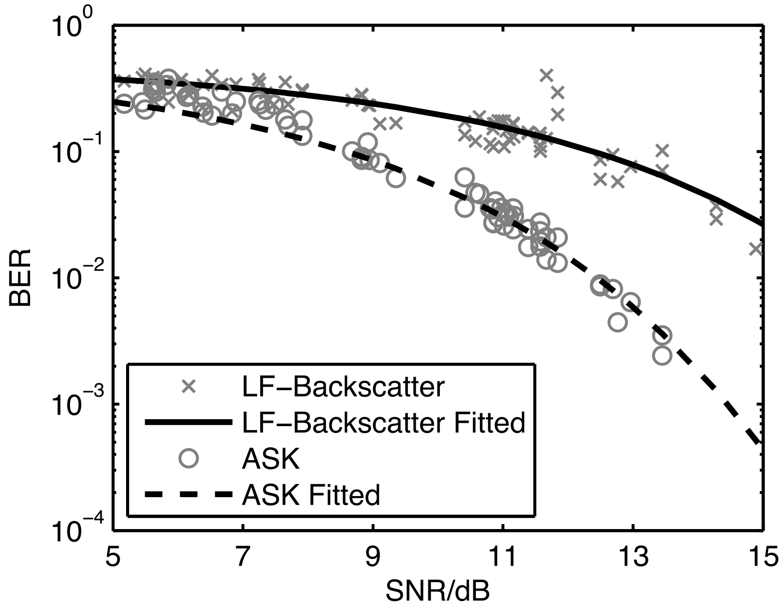
We can observe that LF-Backscatter has stable bits per Joule even when the number of tags increases. In contrast, the energy efficiency of TDMA and Buzz decreases when the number of tags increases because there is too much control message overhead for slot messages, repeated transmissions and dealing with collisions. At the same time, LF Backscatter can be 20x more energy-efficiency than TDMA/Buzz.



What does LF-Backscatter sacrifice?

As with every protocol, LF-Backscatter sacrifices something to get the energy and bandwidth benefits.

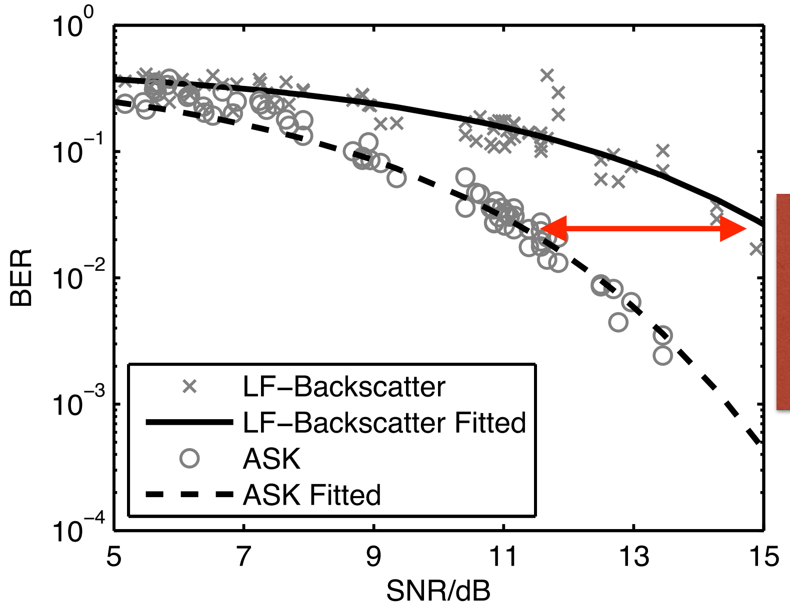
What does LF-Backscatter sacrifice?



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.

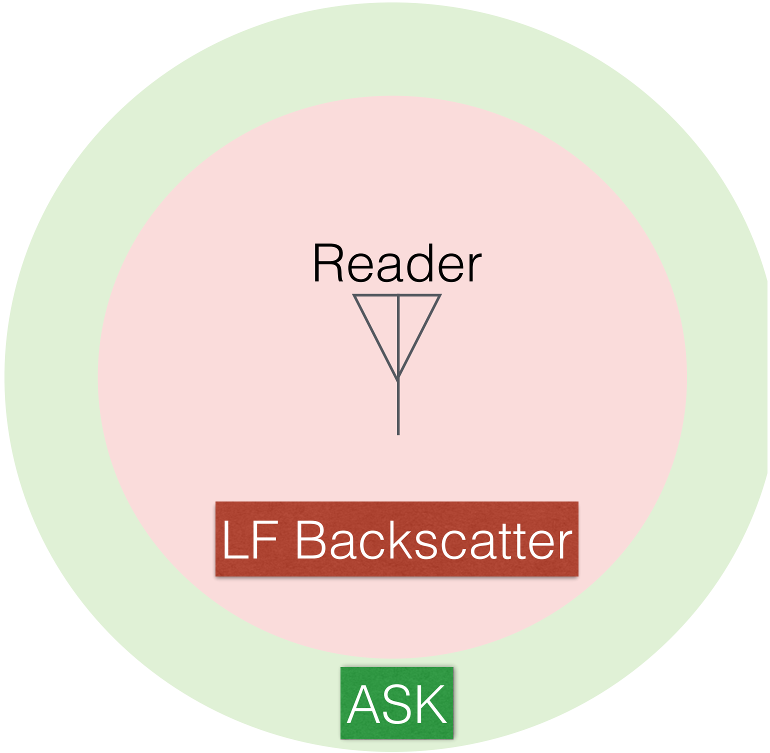
What does LF-Backscatter sacrifice?



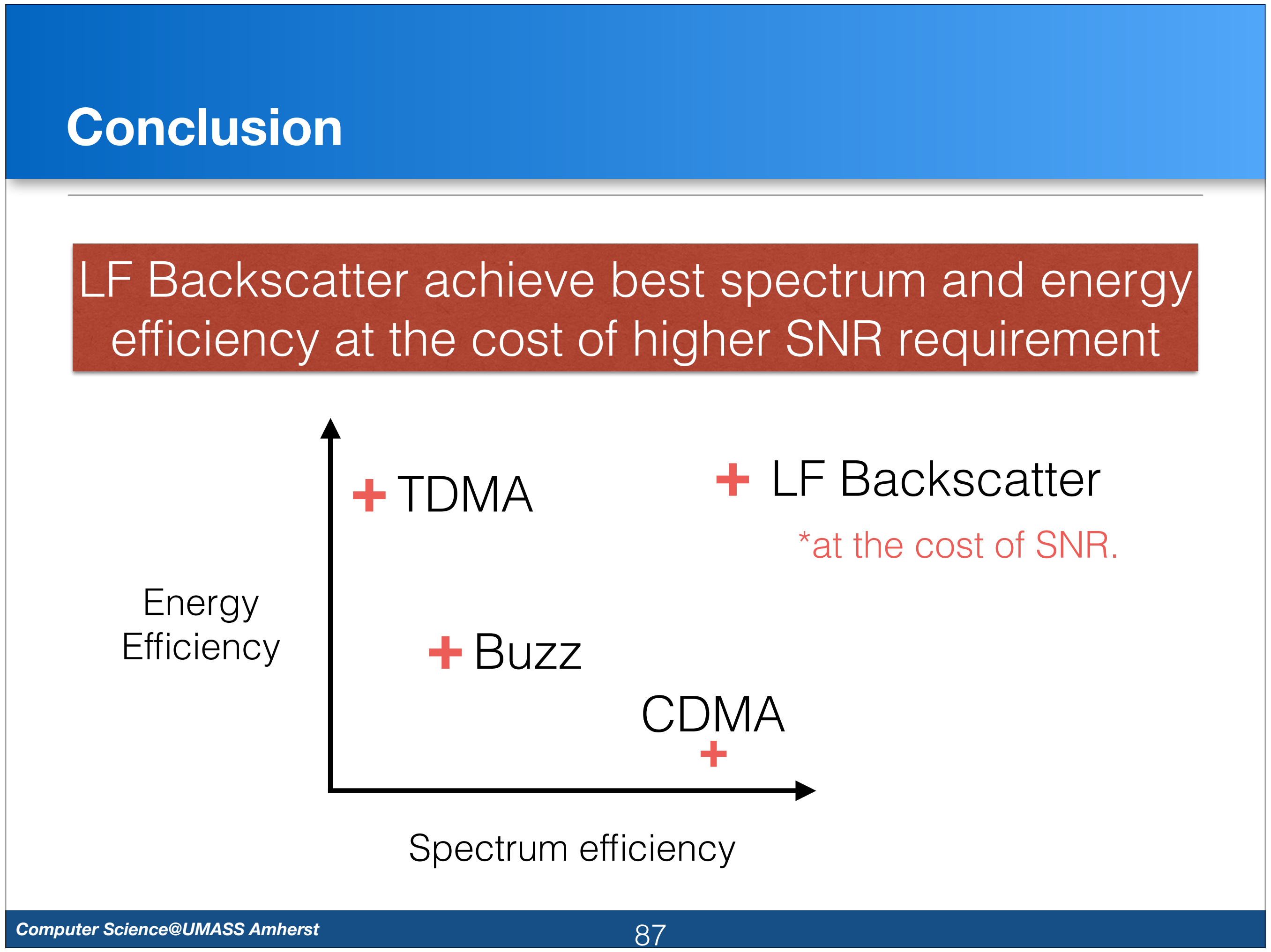
4dB additional SNR for decoding. (20% less range)

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.

What does LF-Backscatter sacrifice?



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.



Conclusion

LF Backscatter achieve best spectrum and energy efficiency at the cost of higher SNR requirement

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.

Conclusion

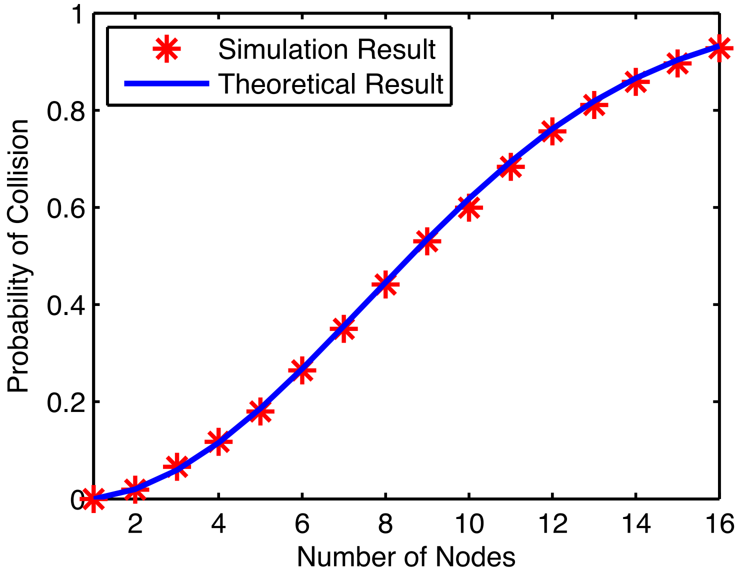
LF Backscatter achieve best power and energy efficiency at the cost of higher SNR requirement

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.

BACKUP SLIDES

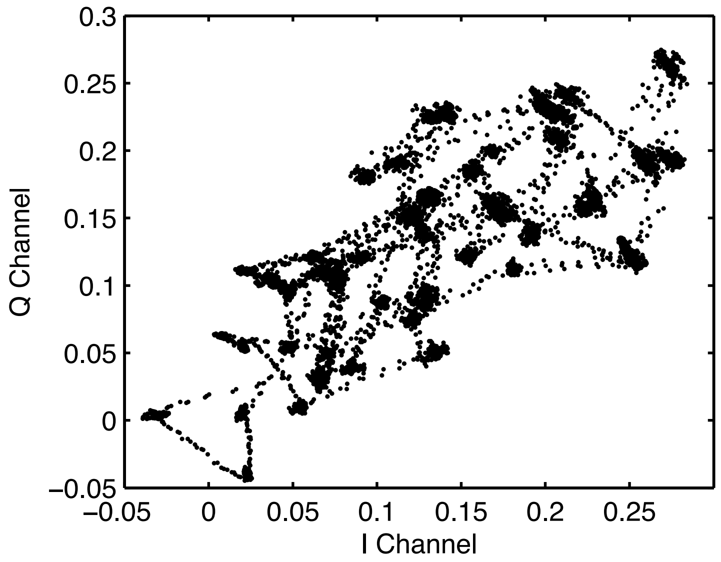
To conclude, LF-Backscatter is a protocol that allows multiple tags to transmit at whatever bitrate they want to. At the heart of our work is the power-bandwidth tradeoff inherent in ultra low-power backscatter-based devices. LF-Backscatter looks at the trade off between throughput, power and SNR and 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.

BACKUP SLIDES



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.

BACKUP SLIDES



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.

BACKUP SLIDES

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.