# X_405082
# Advanced Computer Networks

## Wireless and Mobile

Lin Wang (lin.wang@vu.nl)

Period 2, Fall 2020

VRIJE UNIVERSITEIT AMSTERDAM

# Course outline

## Warm-up

- Fundamentals
- Forwarding and routing
- Network transport

## Data centers

- Data center networking
- Data center transport

## Programmability

- Software defined networking
- Programmable forwarding

## Video

- Video streaming
- Video stream analytics

## Networking and ML

- Networking for ML
- ML for networking

## Mobile computing

- **Wireless and mobile** 👈

# Recap network stack

| | | | | | |
|---|---|---|---|---|---|
| Application | HTTP | RTP | QUIC | FTP | SMTP |
| Transport | | TCP | UDP | MPTCP | |
| Network | | | IPv4/v6 | | |
| Link | | ETH | WLAN | Cellular | |
| Physical | 1000BASE-T | Optical | | RF | Visible light |

# Learning objectives

How can we achieve **battery-free communication**?

Can we **track human** movements **using radio signals**?

# How power small devices?

Power supplies: not flexible
for IoT devices


Battery: hazardous, high maintenance
cost, not suitable for implants

Currently, we have to choose one from these solutions.

# Leveraging existing wireless signals

TV signal

Cellular signal


WiFi signal

These signals **contain energy** and are **widely available** at almost any time and place, despite rains or shine

# Energy harvesting

Recent work shows that we can harvest energy from ambient radio signals up to 10s of uW

Enough for computation and sensing

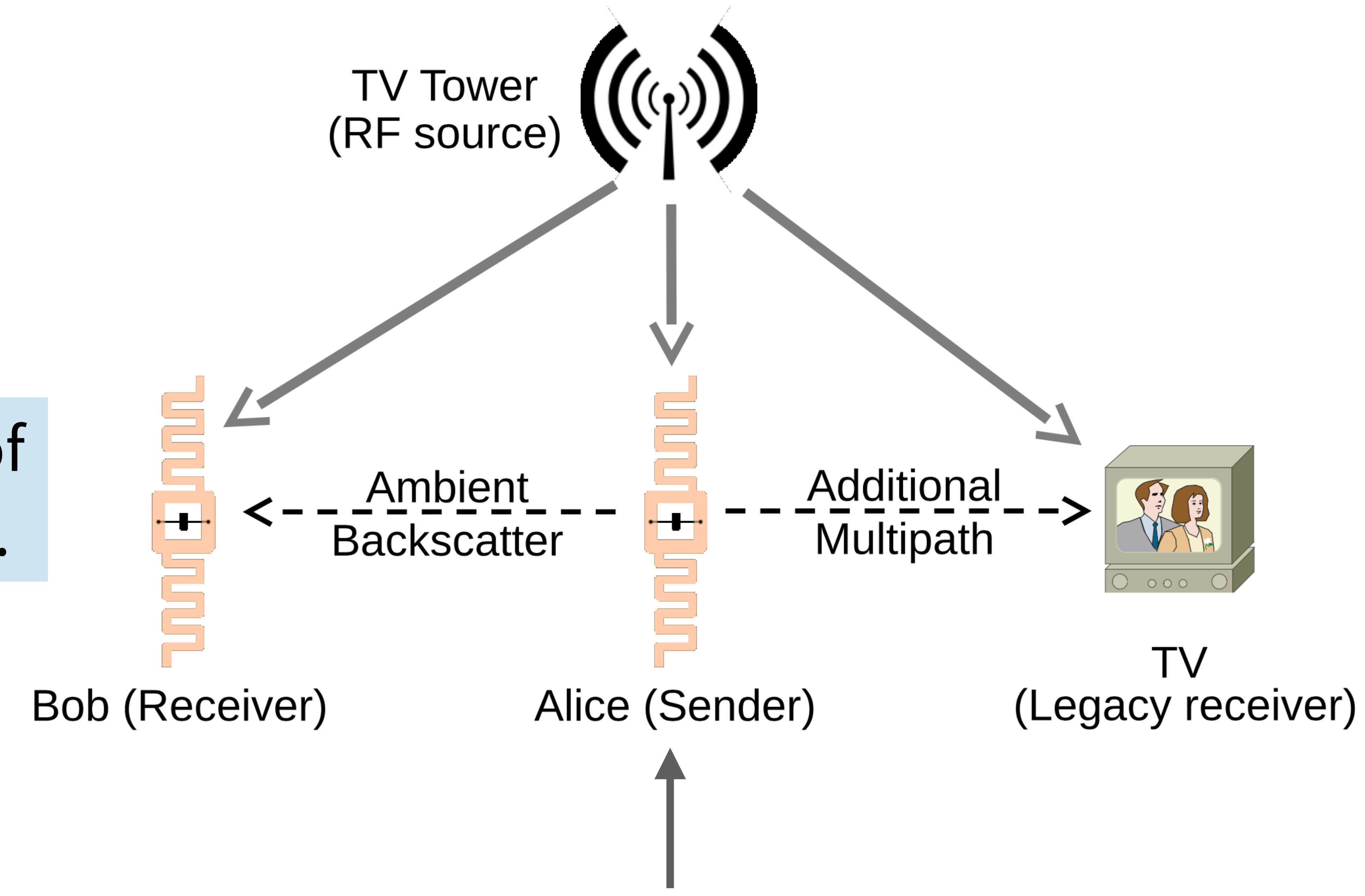
Orders of magnitude less power than needed for radio communication

- Generating radio signals is expensive
- Could duty cycle: limits interactive applications

Can we communicate without either device generating radio signals?

# Ambient backscatter

Use existing signals instead of generating our own



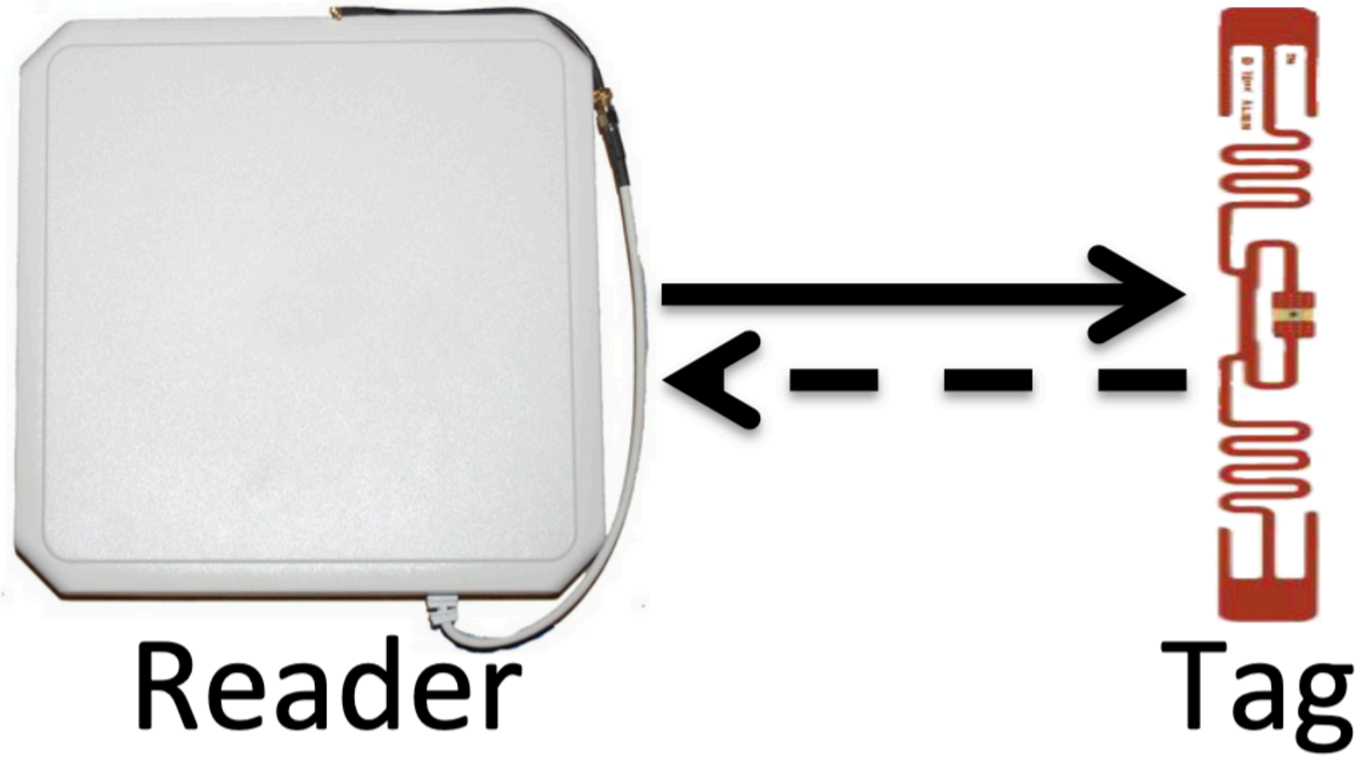

Works with only 5% of the harvested power.

'0': absorb TV signals, '1': reflect TV signals

# RFID (radio frequency identification)



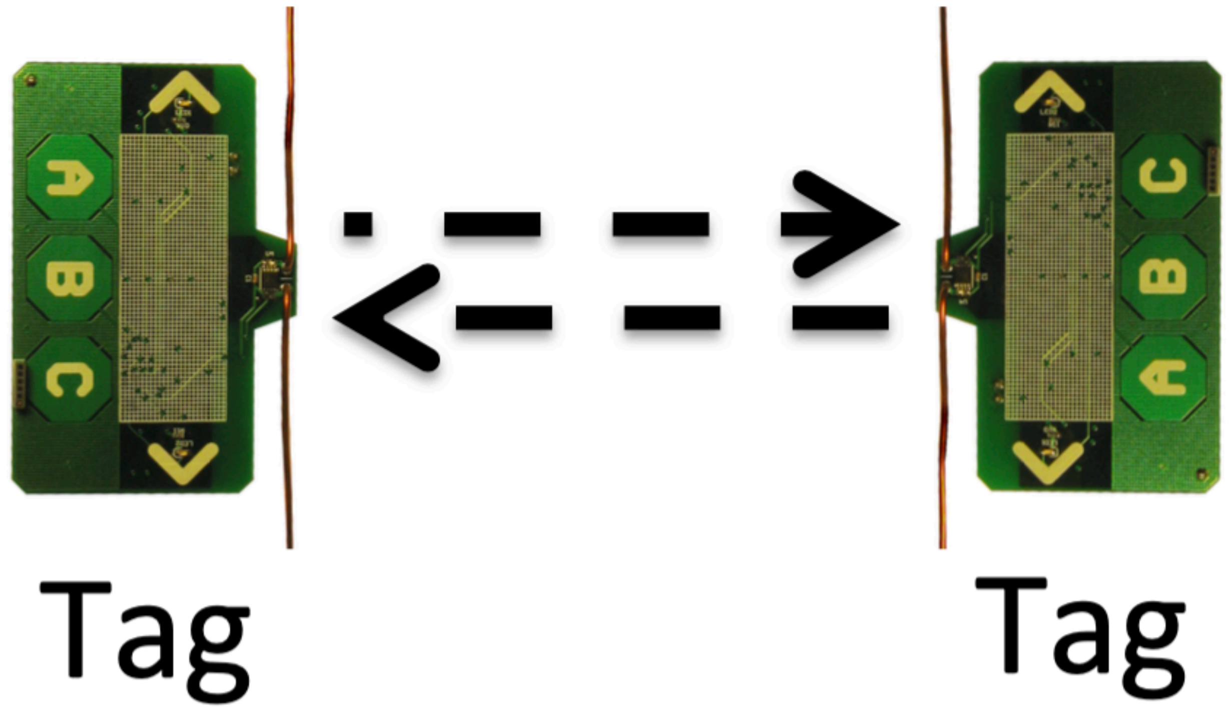
Reader sends constant wave




Receive chain:
100s of mW

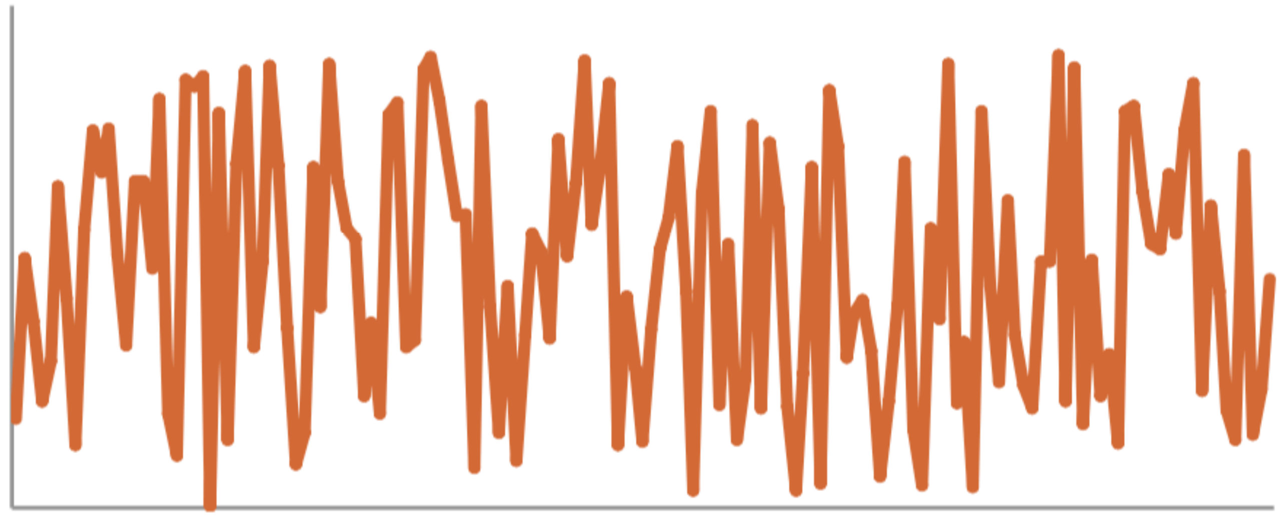
Signal is centrally coordinated by the reader

# Properties of ambient backscatter




Receive chain: 0.5uW

Requires distributed MAC (medium access control) protocol like WiFi



Uses uncontrollable signals

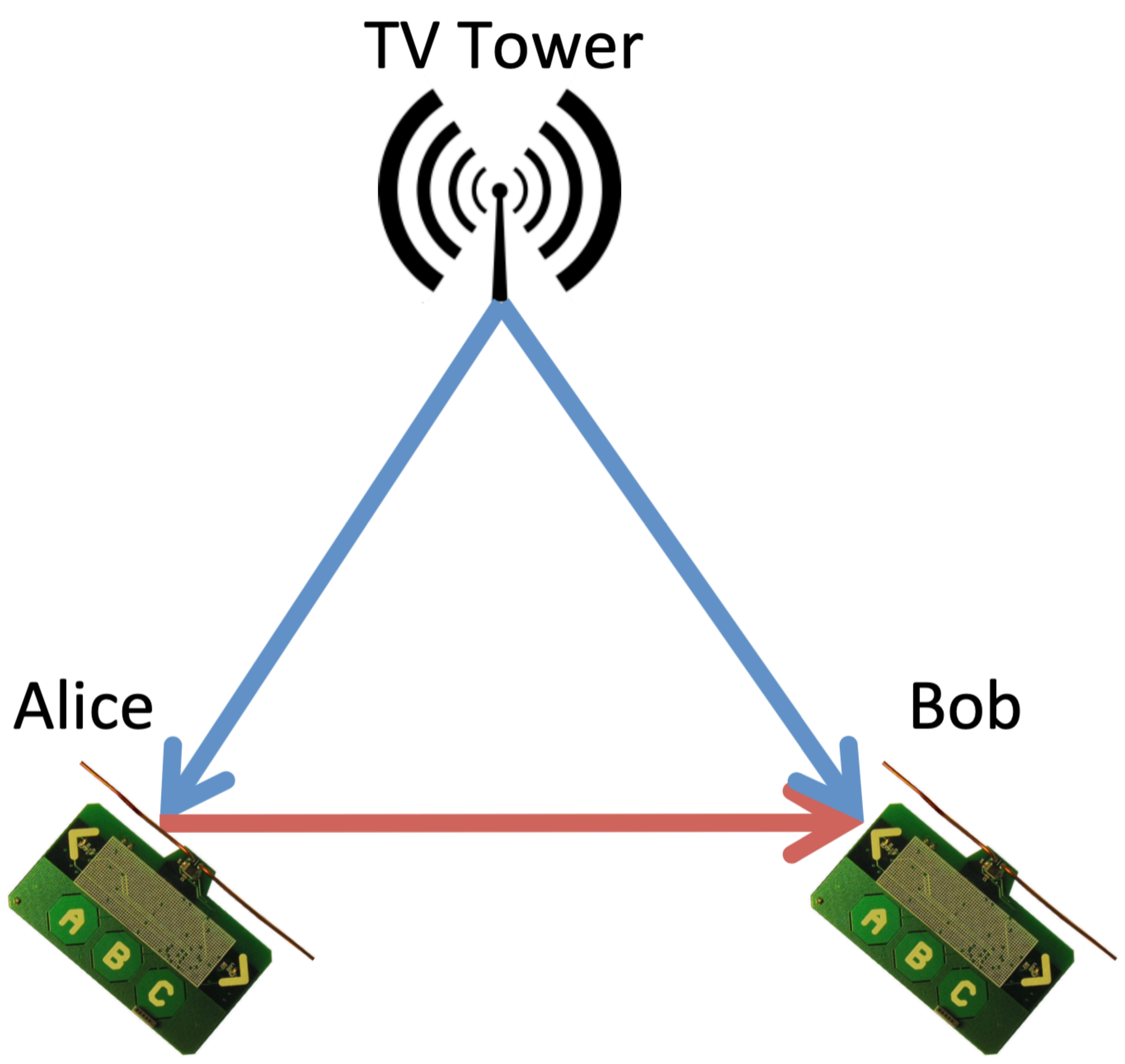
# Challenges with ambient backscatter

**Extracting backscattered signals** from ambient signals we do not control

**Decoding** on a battery-free device

Designing **distributed MAC** for battery-free devices

# Challenge 1: extracting backscattered signals



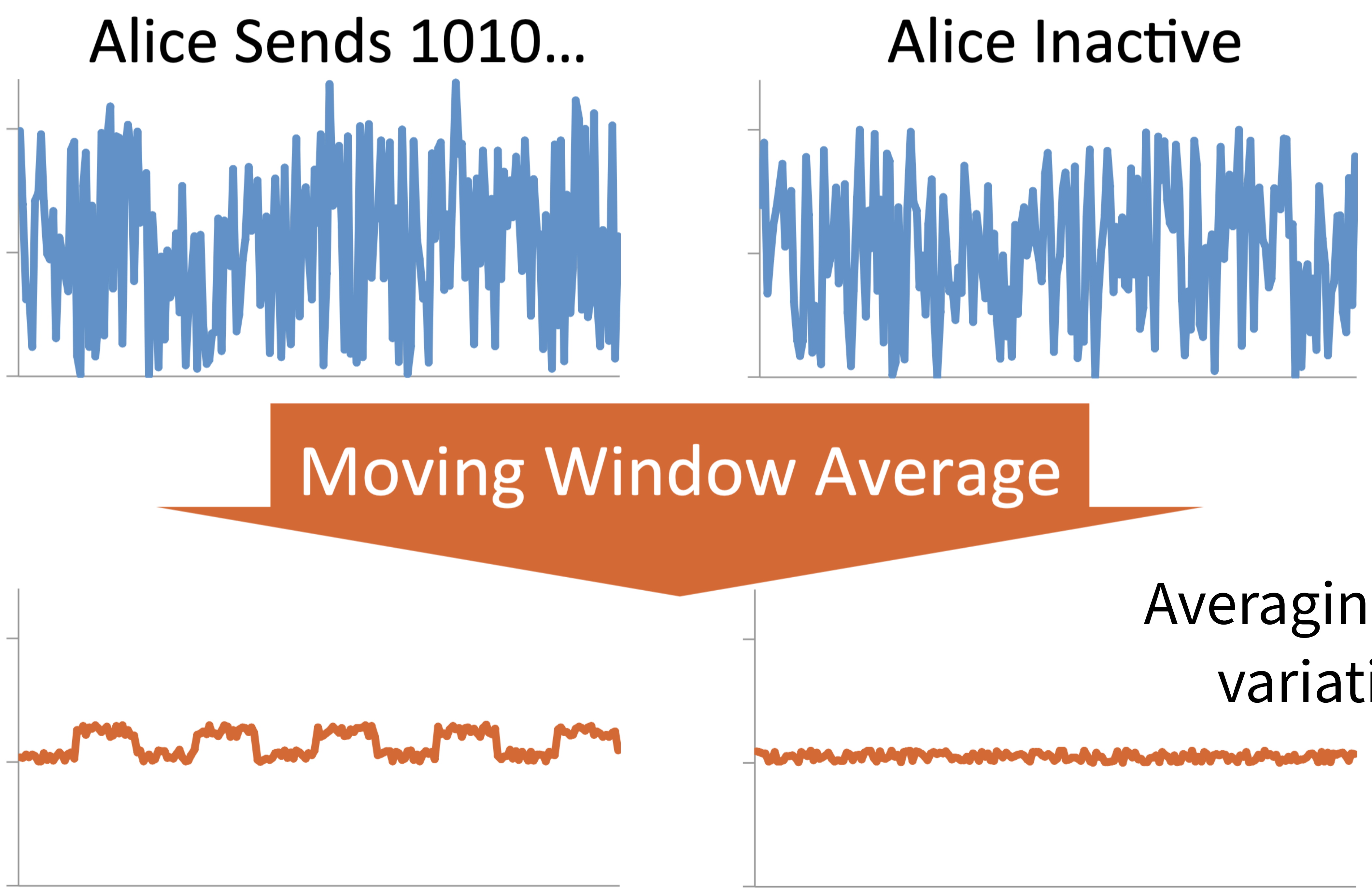

'0': Alice absorbs the TV signal
**At Bob:** only TV signal

'1': Alice reflects the TV signal
**At Bob:** TV signal + weak reflection

Observation: Alice's reflection changes the average amplitude at Bob

Make sure the ambient TV signals change at a faster rate than the backscatter communication.

# Solution: detect changes in average amplitude

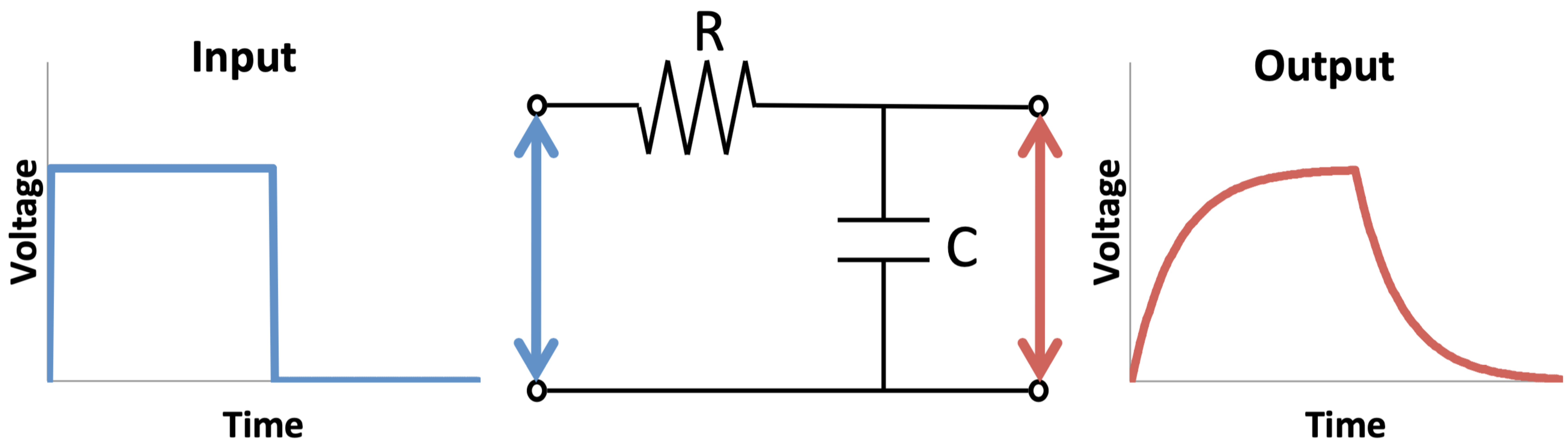



Averaging smoothes out the variation in TV signals.

Averaging would be easy if we have digital samples on the signals. However, it requires **analog to digital converters** which are typically **power-hungry**.

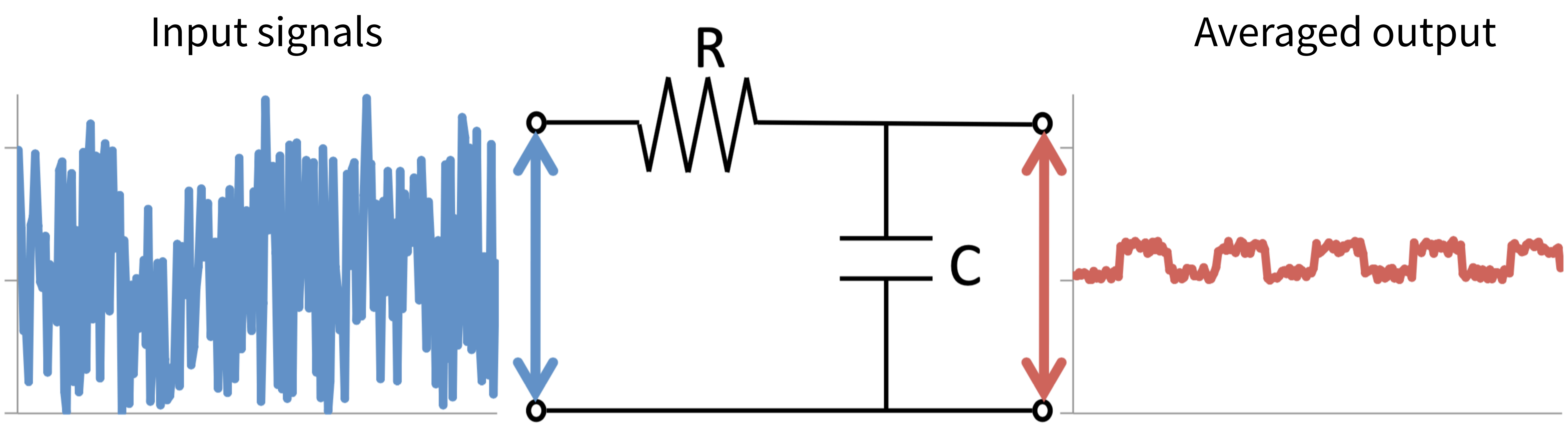
# Challenge 2: decoding on a battery-free device

**Time for some basic physics:** capacitors slowly charges/
discharges when voltage is applied/removed




The resistive-capacitive (RC) circuits provide a cheap, analog, exponential moving average.

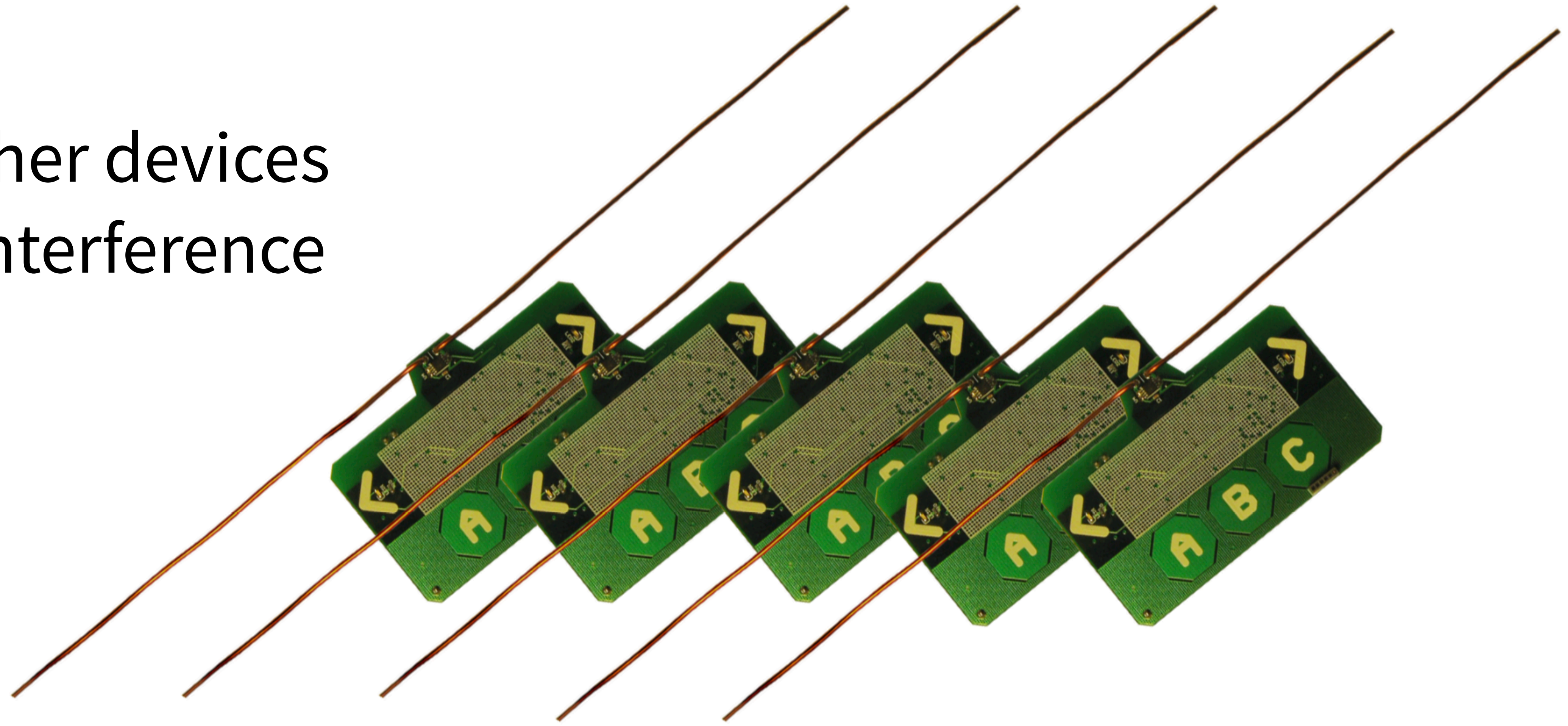
# Average with RC circuits

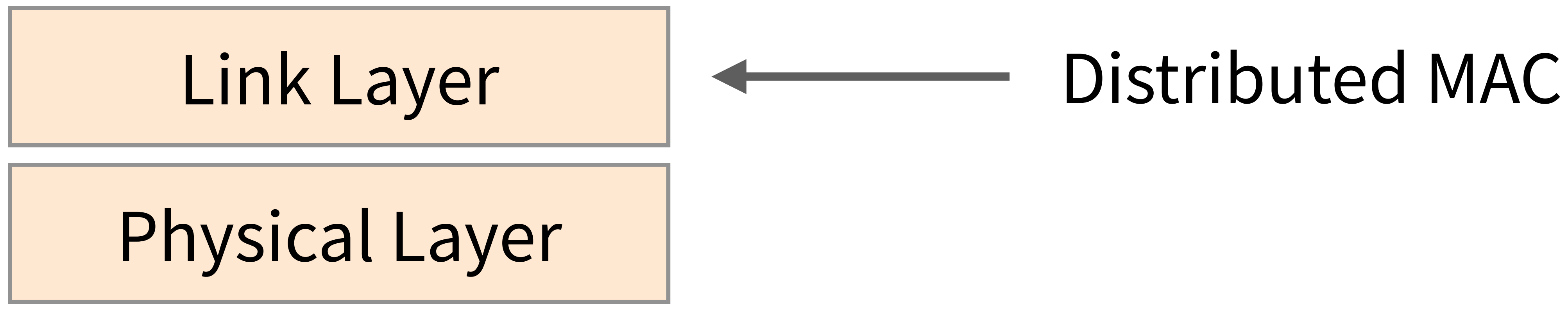



By picking the right RC values, we can selectively filter out the high TV frequencies.

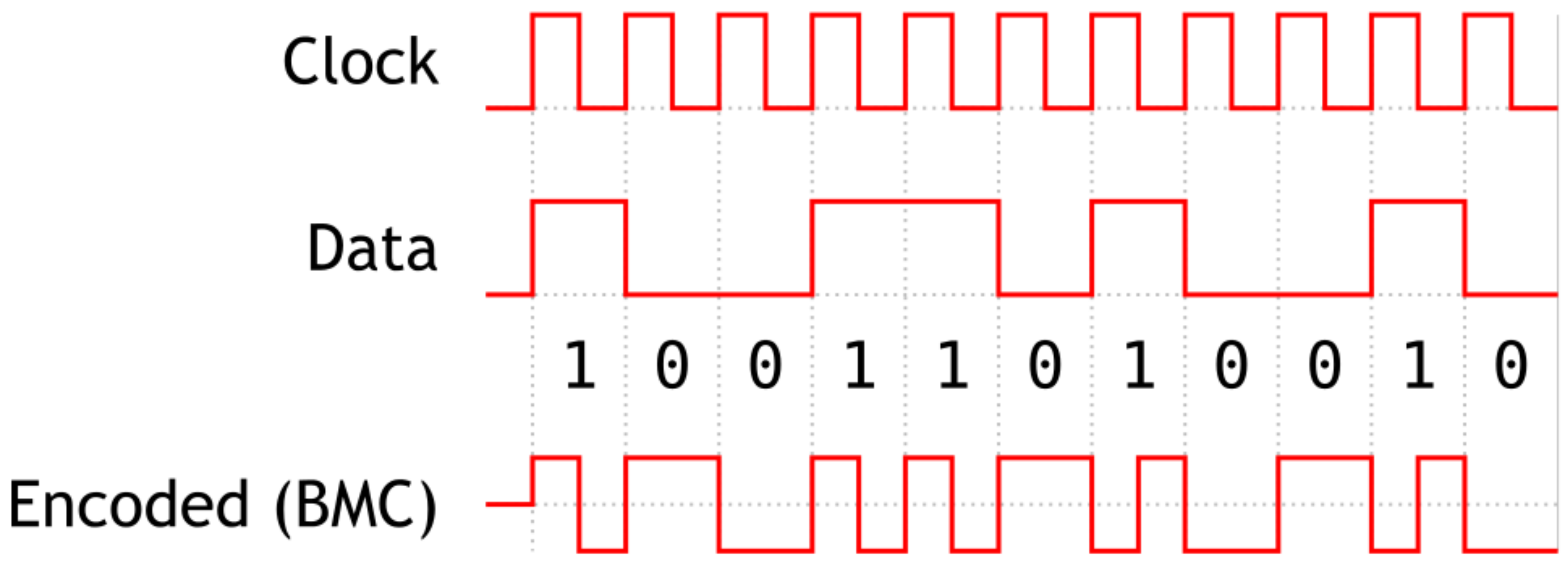
# Challenge 3: distributed MAC

Any device may communicate to other devices
using the same shared medium → interference





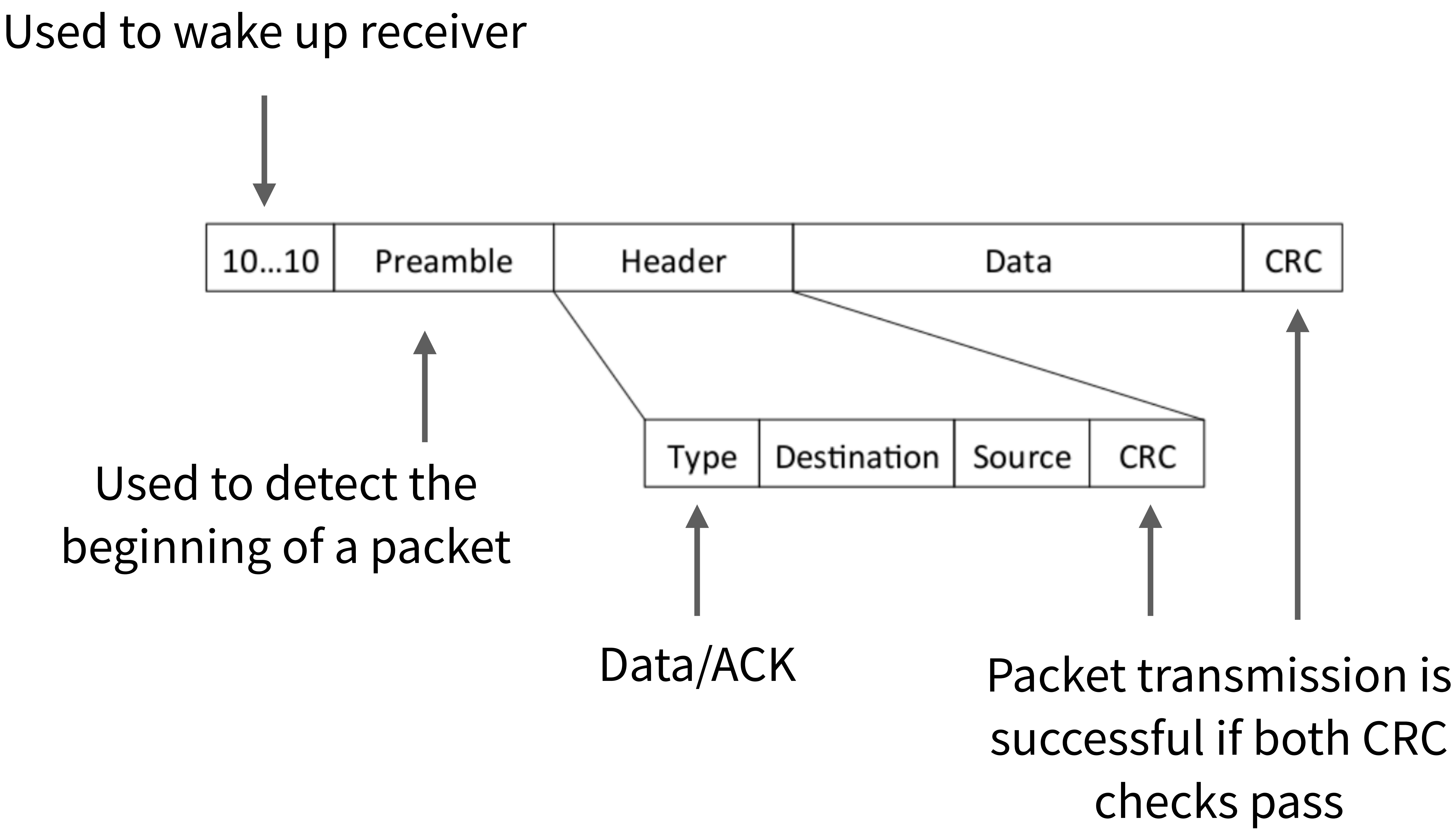
# Modulation and bit encoding




Equal number of high and low

**FM0 coding:** data is encoded with the transition of power levels, not the absolute power level (mid-bit transition for '1' and no transition for '0')

# Packet format



# CSMA (carrier-sense multiple access)

CSMA uses carrier sense, i.e., energy detection

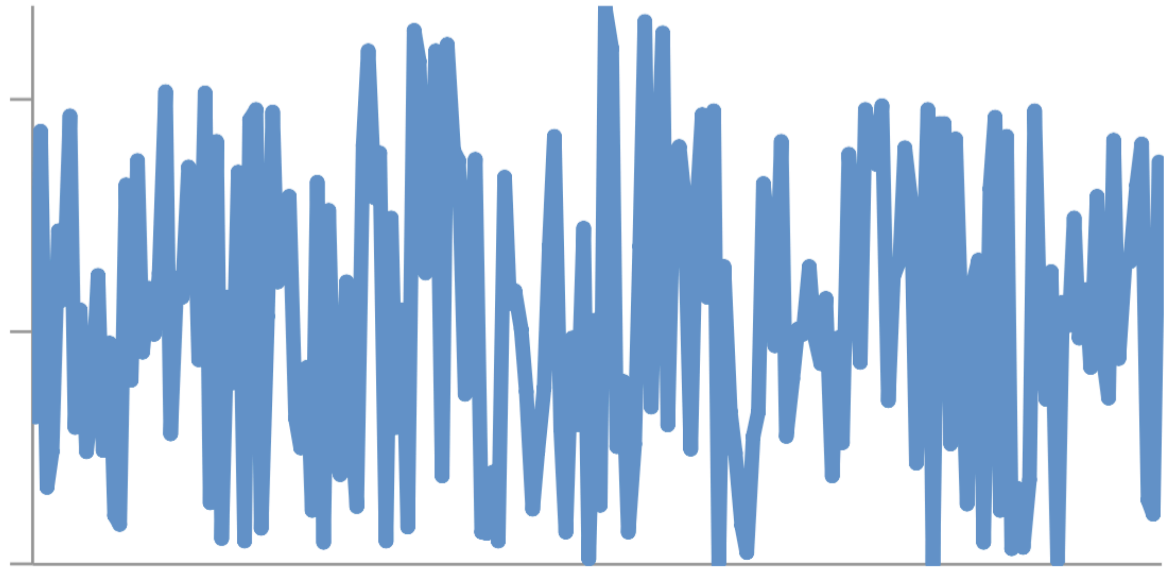
Battery-free devices do not have energy levels

- Requires power-hungry Analog-Digital-Converters (ADCs)
- Challenge: energy detection without access to the energy levels

Use **bit transitions in FM0 coding** as proxy for energy detection

No backscatter: all '0's or all '1's

Backscatter: equal number of '0's and '1's



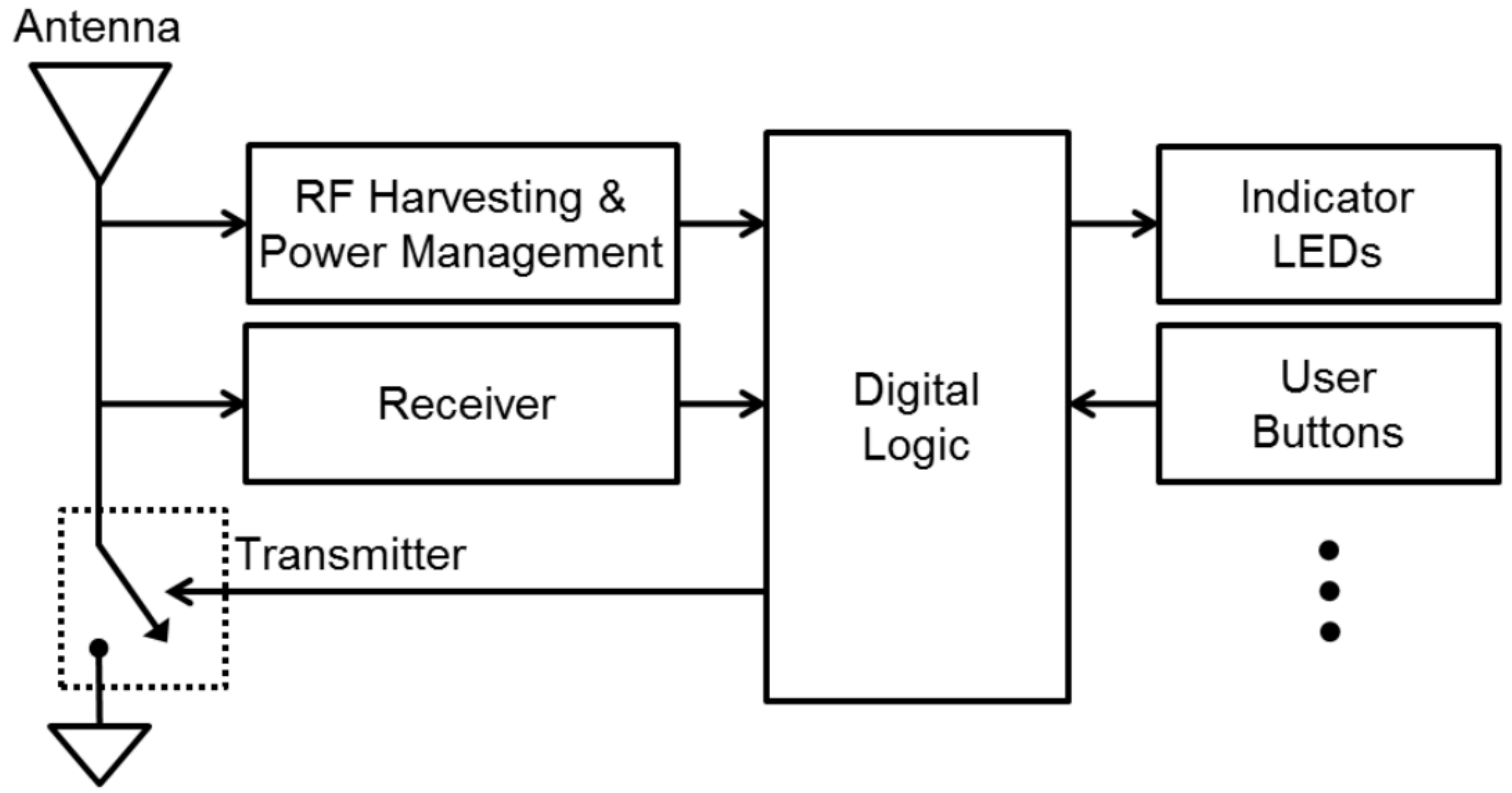
# Evaluation

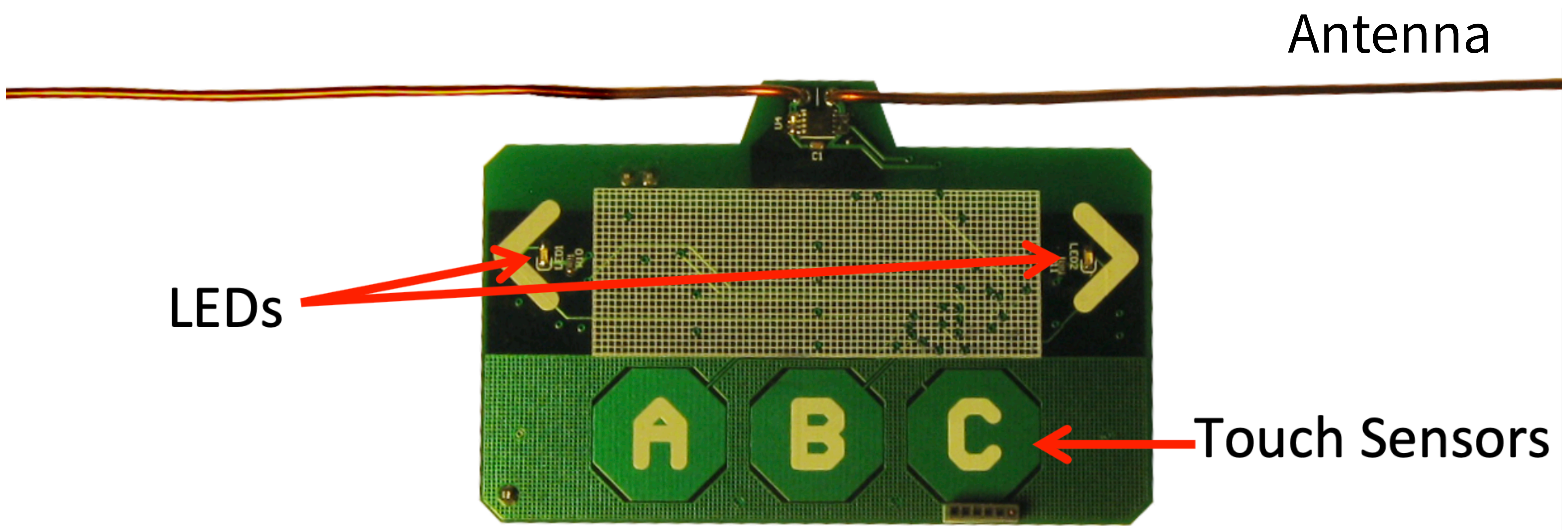
TV Tower

10.5km from the TV tower

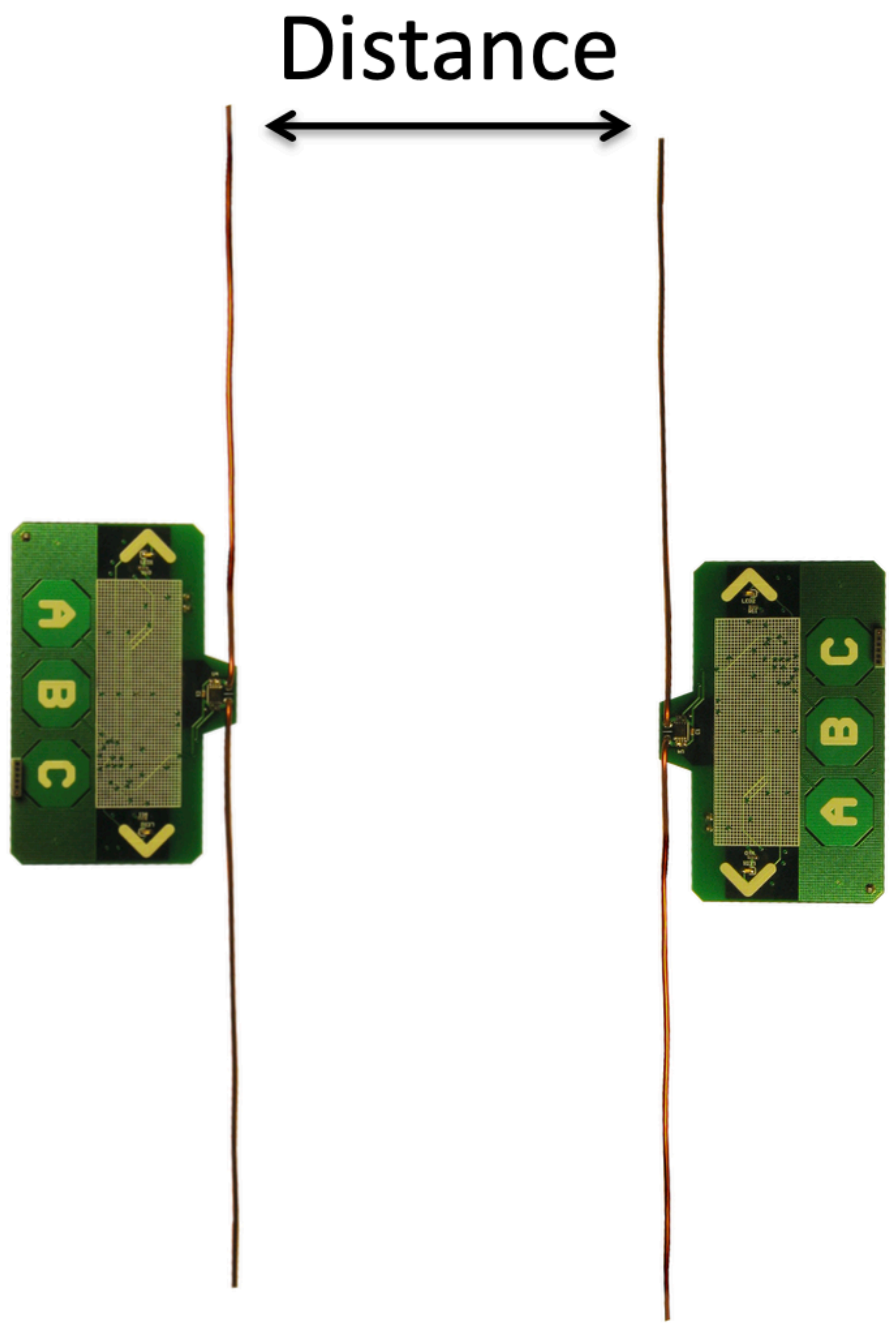
## Prototype using off-the-shelf components

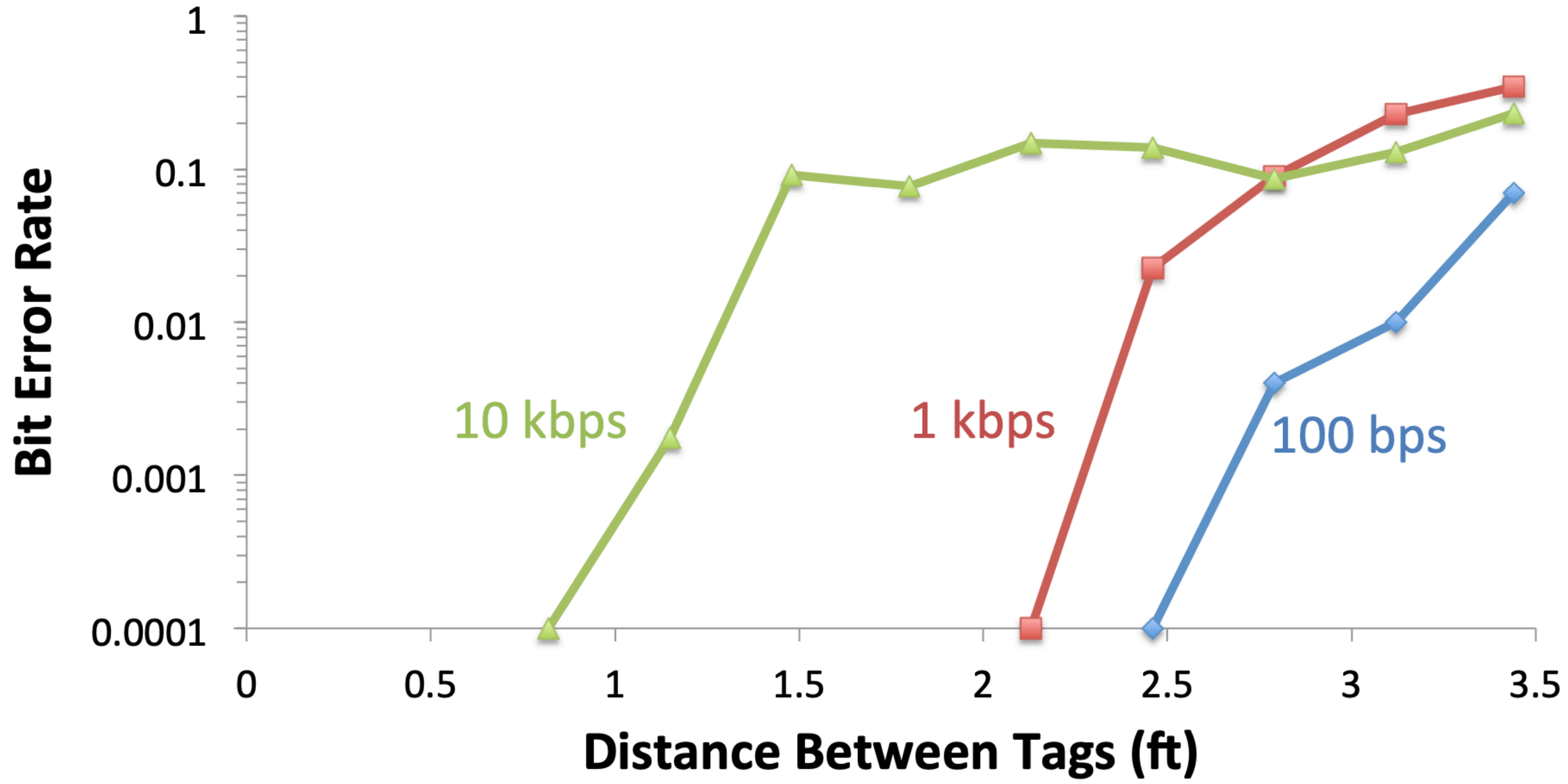
- Battery-free
- Harvests and backscatters TV signals at 539 MHz
- Microcontroller performs computation



# Results



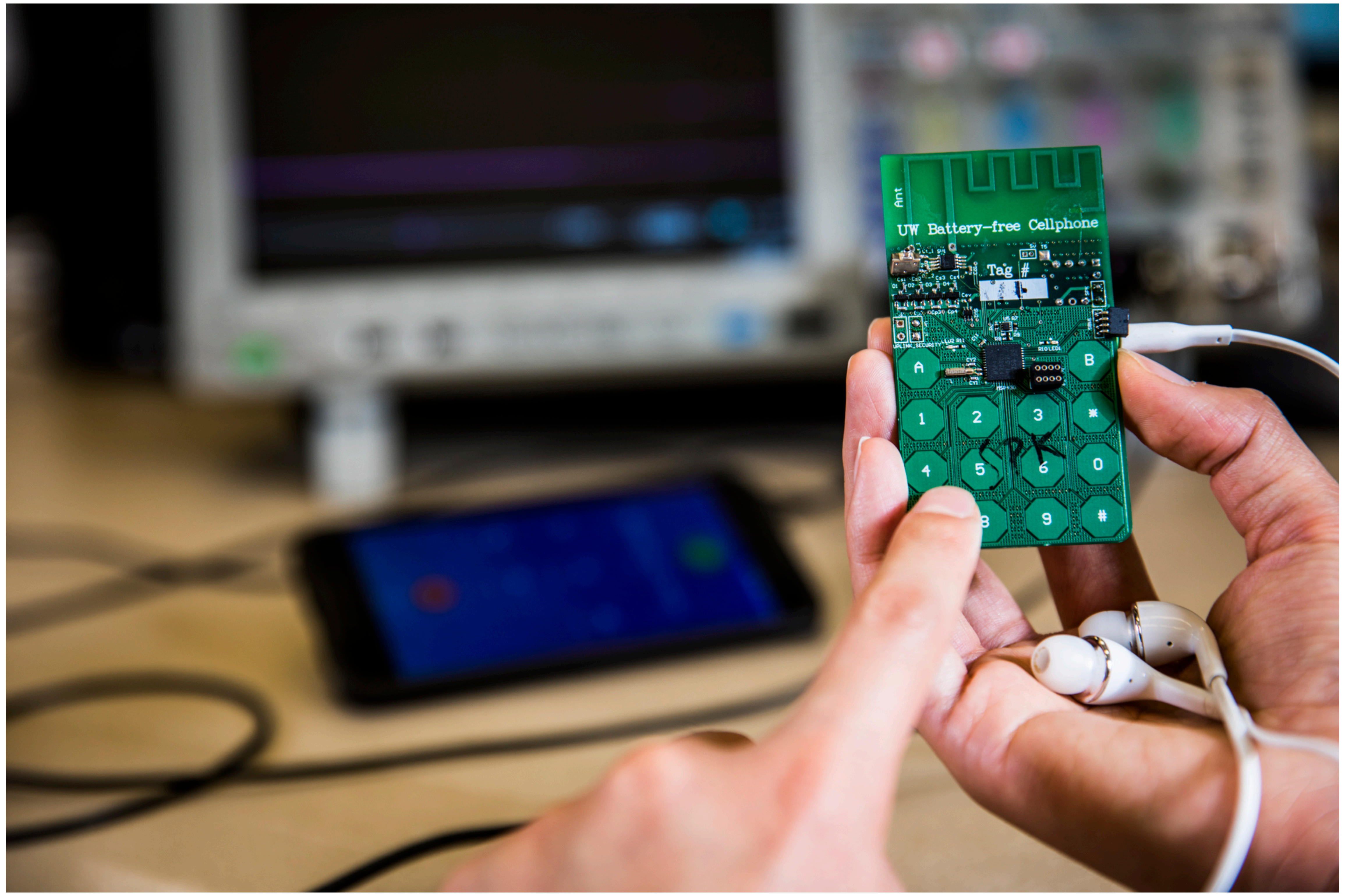

The results confirm the feasibility of ambient backscatter.

# Applications


Grocery store application where tags are used to identify misplaced items in grocery stores or in warehouses


Smart card application where cards can exchange information directly without involving a third party

# Battery-free cellphone



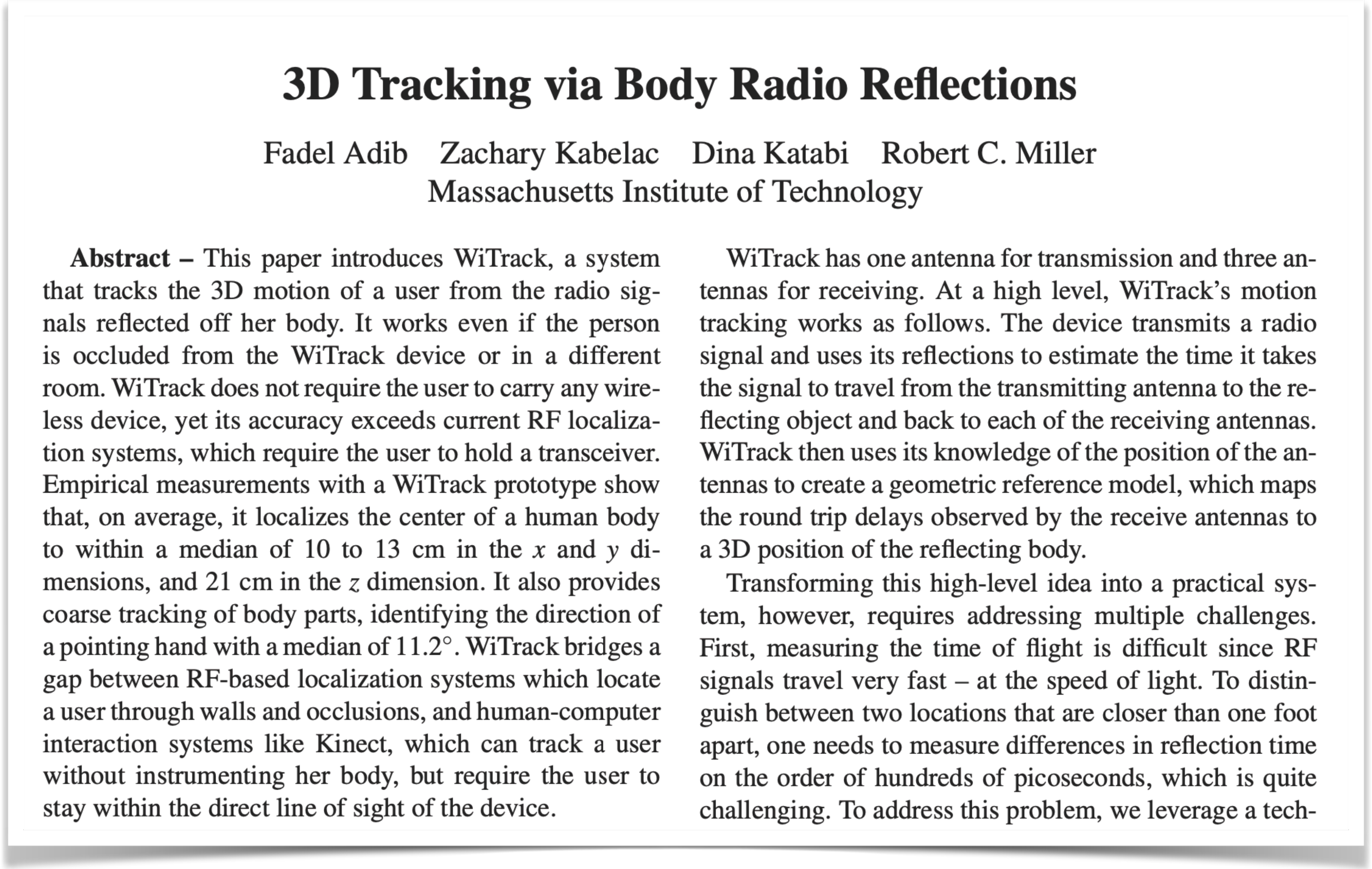

https://batteryfreephone.cs.washington.edu

## Can we see through the walls with wireless signals?

# WiTrack

## Tracking with WiFi signals

- Centimeter-scale motion tracking using only radio reflections off the human body
- Works behind walls and does not require person to hold any device



**3D Tracking via Body Radio Reflections**

Fadel Adib Zachary Kabelac Dina Katabi Robert C. Miller
Massachusetts Institute of Technology

**Abstract –** This paper introduces WiTrack, a system that tracks the 3D motion of a user from the radio signals reflected off her body. It works even if the person is occluded from the WiTrack device or in a different room. WiTrack does not require the user to carry any wireless device, yet its accuracy exceeds current RF localization systems, which require the user to hold a transceiver. Empirical measurements with a WiTrack prototype show that, on average, it localizes the center of a human body to within a median of 10 to 13 cm in the $x$ and $y$ dimensions, and 21 cm in the $z$ dimension. It also provides coarse tracking of body parts, identifying the direction of a pointing hand with a median of 11.2°. WiTrack bridges a gap between RF-based localization systems which locate a user through walls and occlusions, and human-computer interaction systems like Kinect, which can track a user without instrumenting her body, but require the user to stay within the direct line of sight of the device.

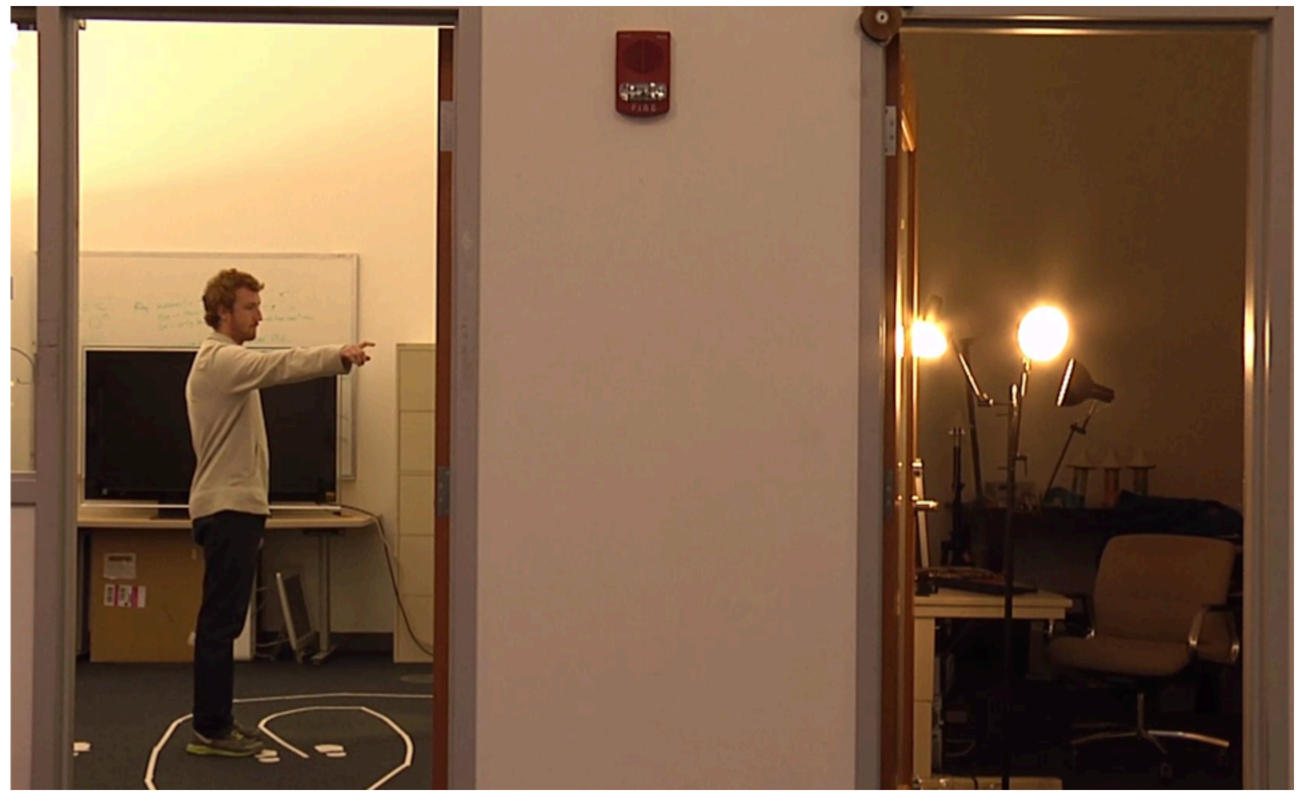
WiTrack has one antenna for transmission and three antennas for receiving. At a high level, WiTrack's motion tracking works as follows. The device transmits a radio signal and uses its reflections to estimate the time it takes the signal to travel from the transmitting antenna to the reflecting object and back to each of the receiving antennas. WiTrack then uses its knowledge of the position of the antennas to create a geometric reference model, which maps the round trip delays observed by the receive antennas to a 3D position of the reflecting body.

Transforming this high-level idea into a practical system, however, requires addressing multiple challenges. First, measuring the time of flight is difficult since RF signals travel very fast – at the speed of light. To distinguish between two locations that are closer than one foot apart, one needs to measure differences in reflection time on the order of hundreds of picoseconds, which is quite challenging. To address this problem, we leverage a tech-

USENIX NSDI 2014

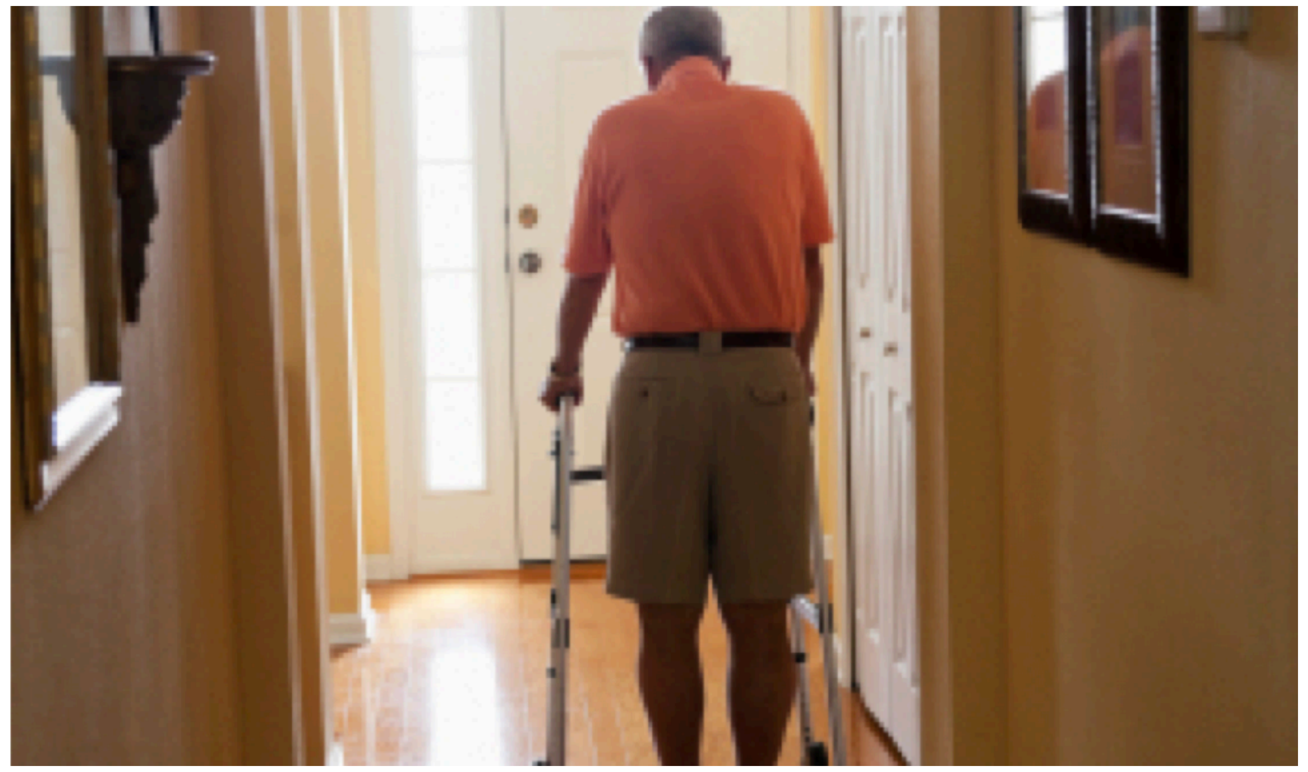
# Many applications

Gaming



Gesture control

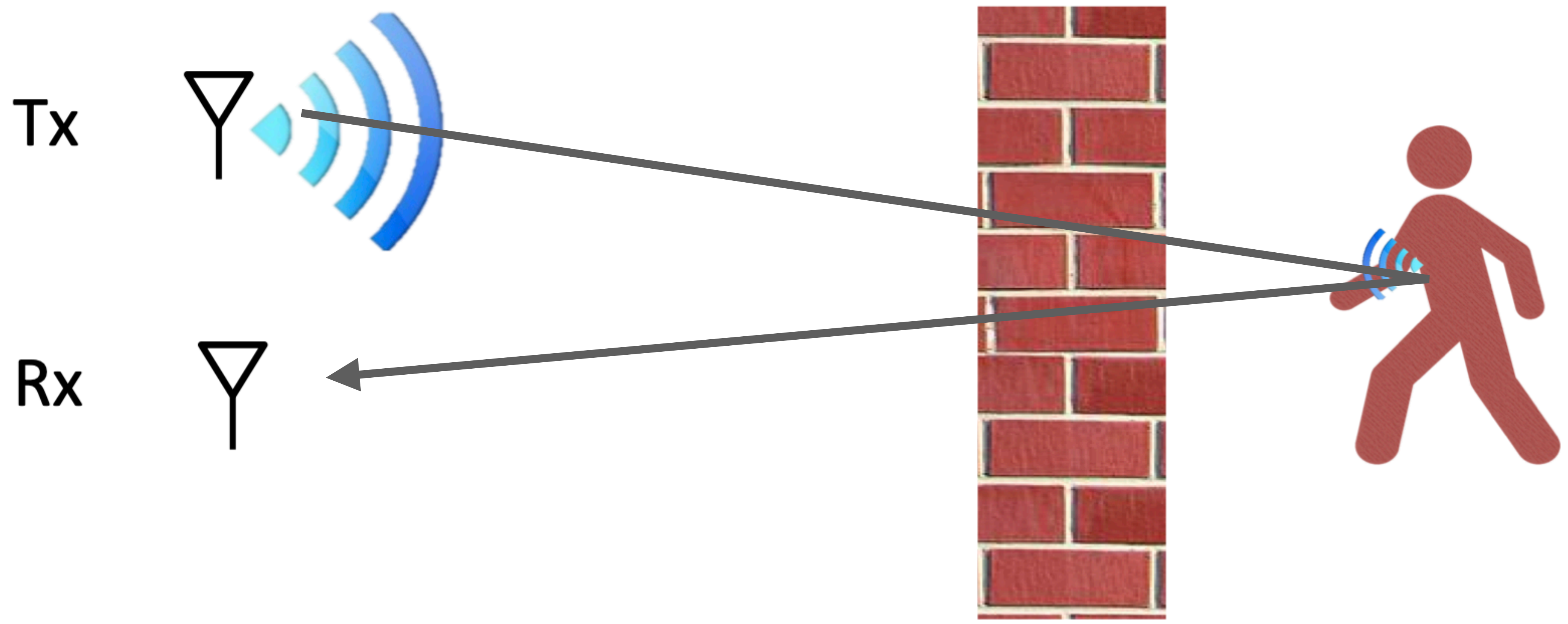
First responders



Elderly monitoring

# How to measure distance

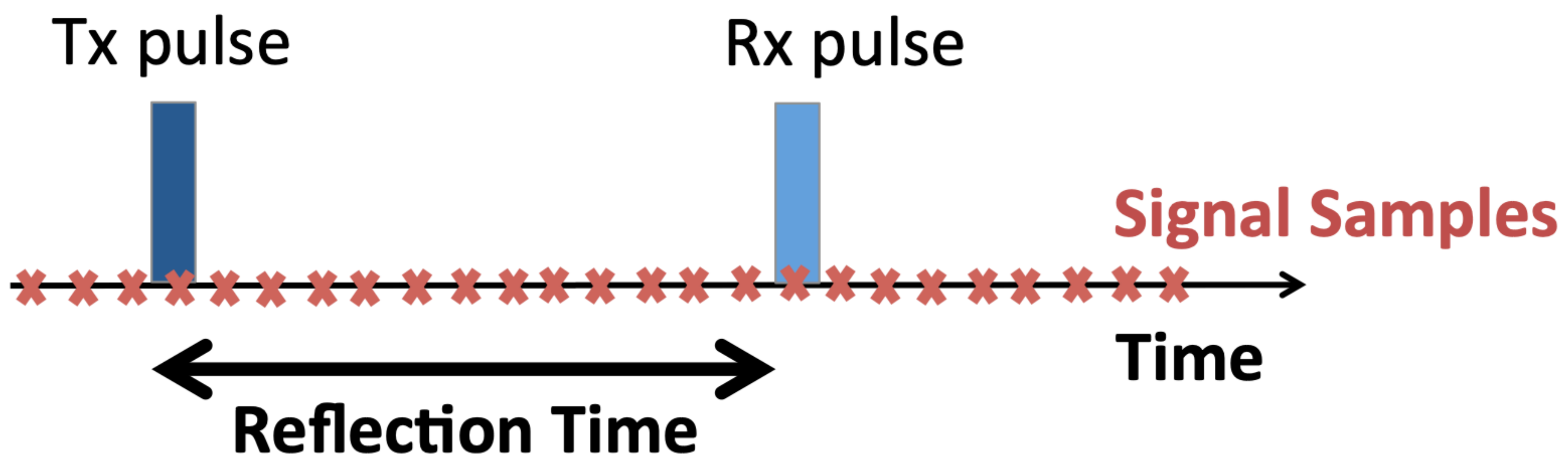
Distance = **reflection time** x speed of light




How to measure reflection time?

# Measuring reflection time

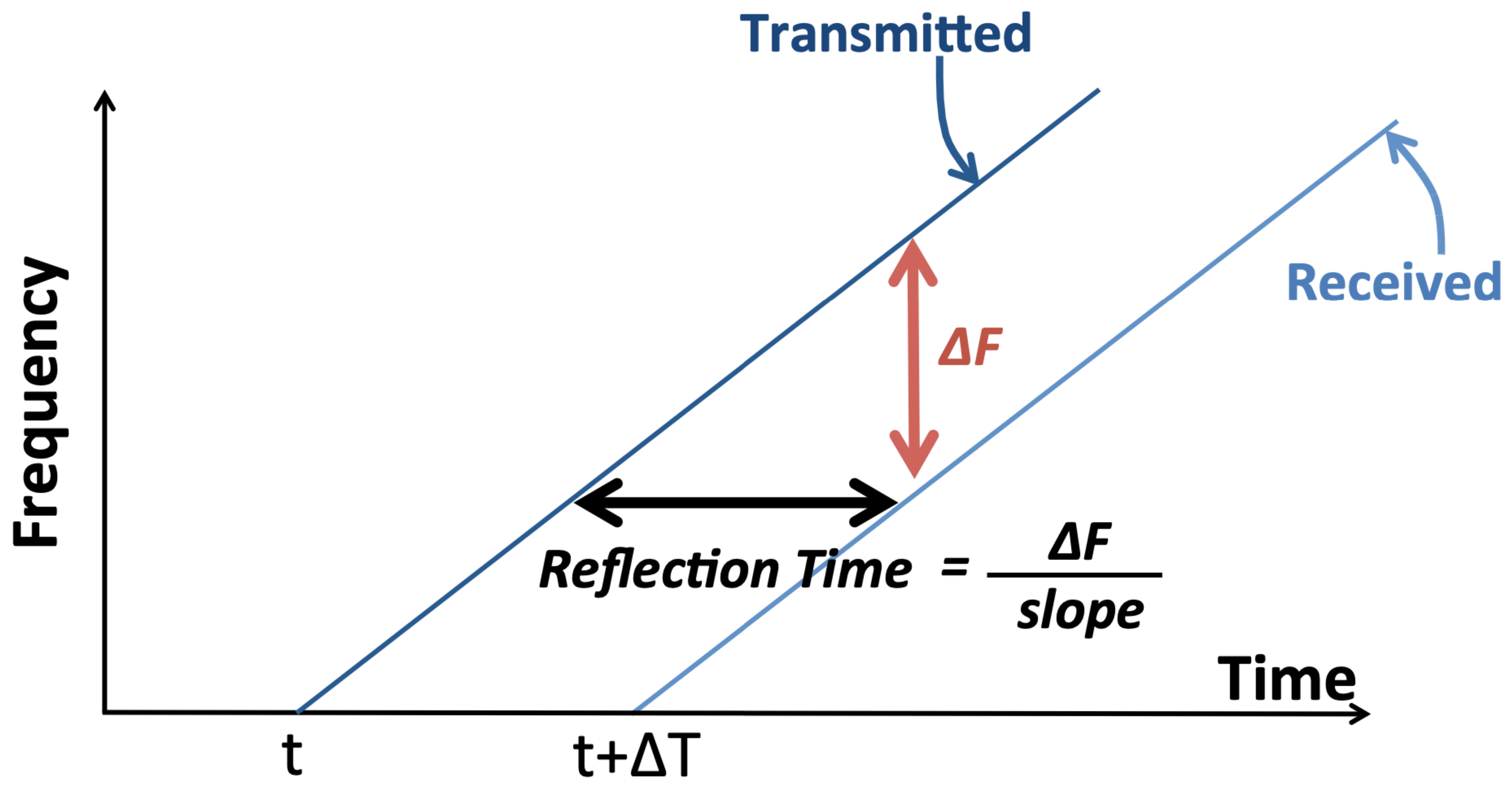
**Option 1:** transmit short pulses and listen for echo




Problem: capturing the pulse requires sub-nanosecond sampling, which is expensive and has high noise → impractical for the applications

# Measuring reflection time

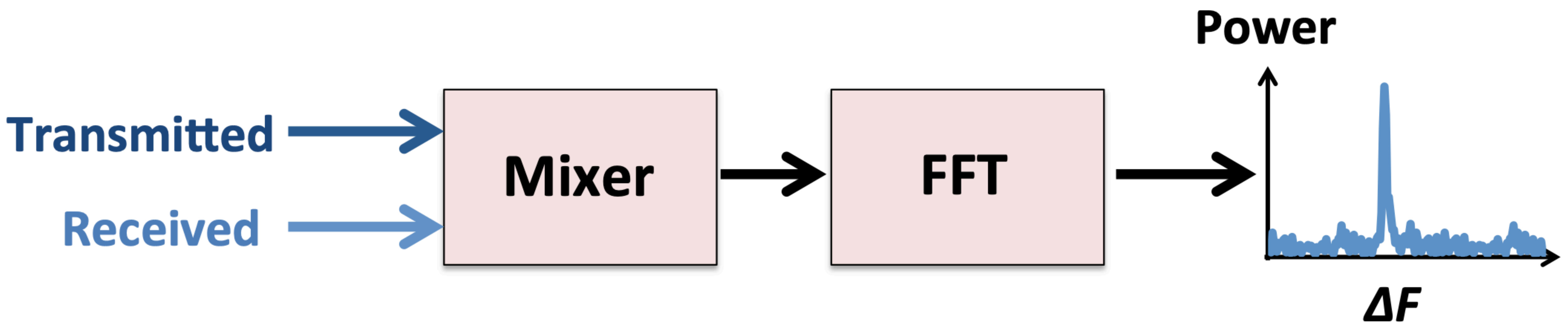
**Option 2:** frequency modulated carrier wave (FMCW) - widely used in radar systems




How to measure ΔF?

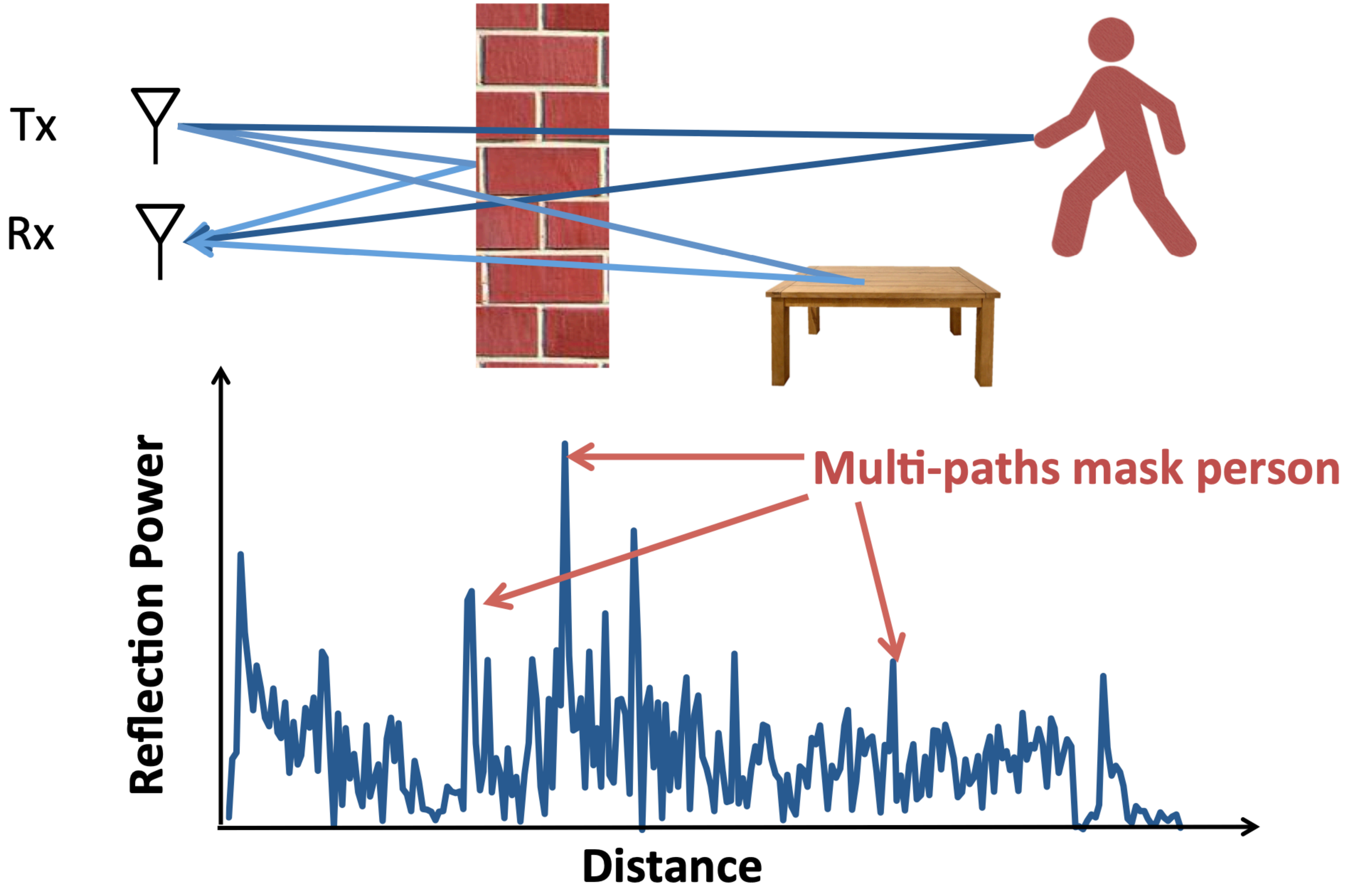
# Measuring ΔF

Subtracting frequencies is easily done using a mixer (low-power, cheap), e.g., removing carrier in WiFi (22MHz from 2.4GHz)

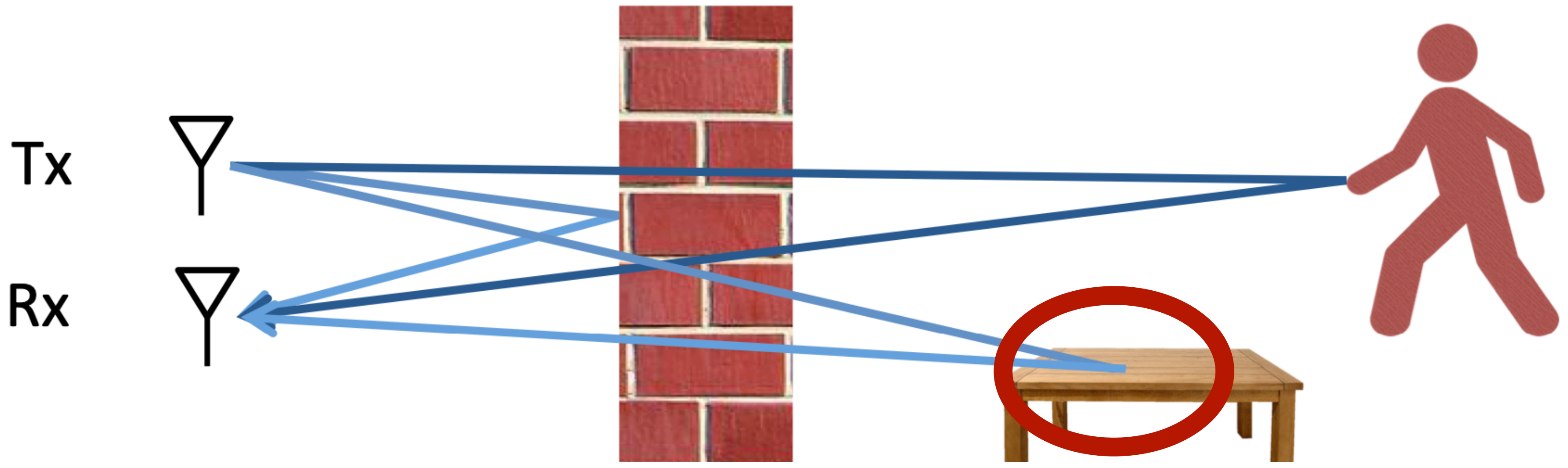



We can obtain the distance via: ΔF → reflection time → distance

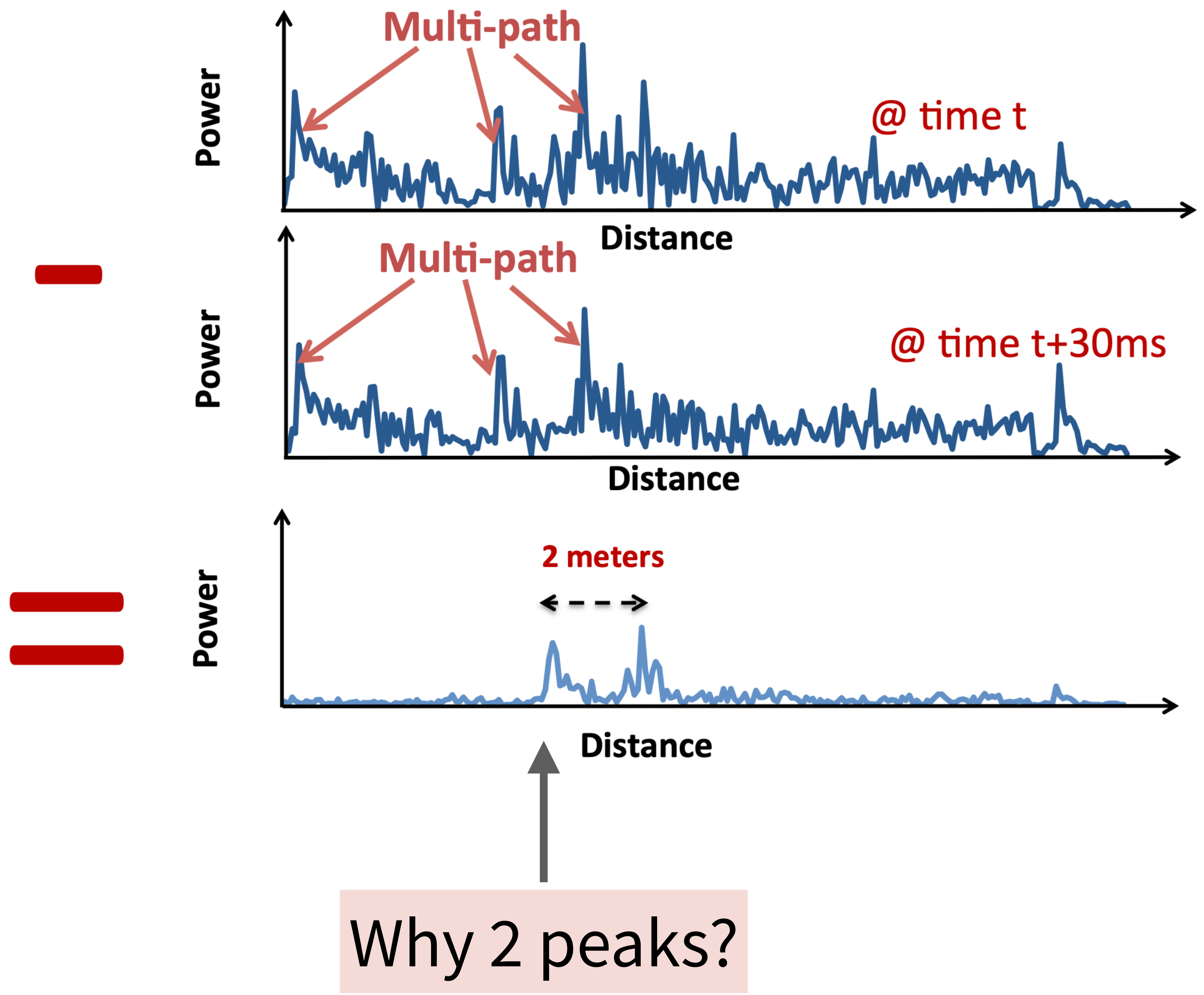
# How to deal with many reflections



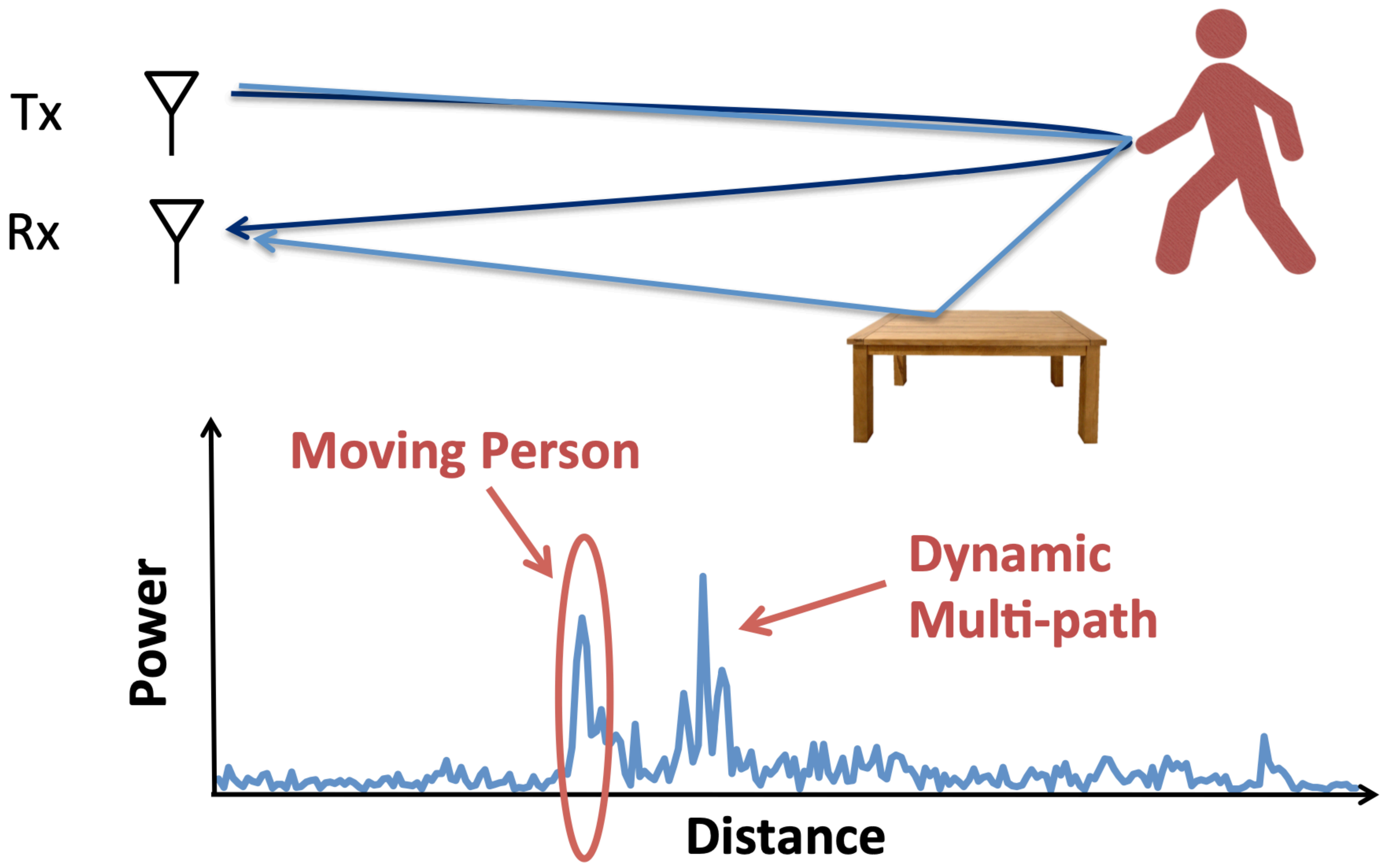
# Removing reflections by static objects




Reflections from static objects are constant, which can be removed by subtracting the signals measured at different times

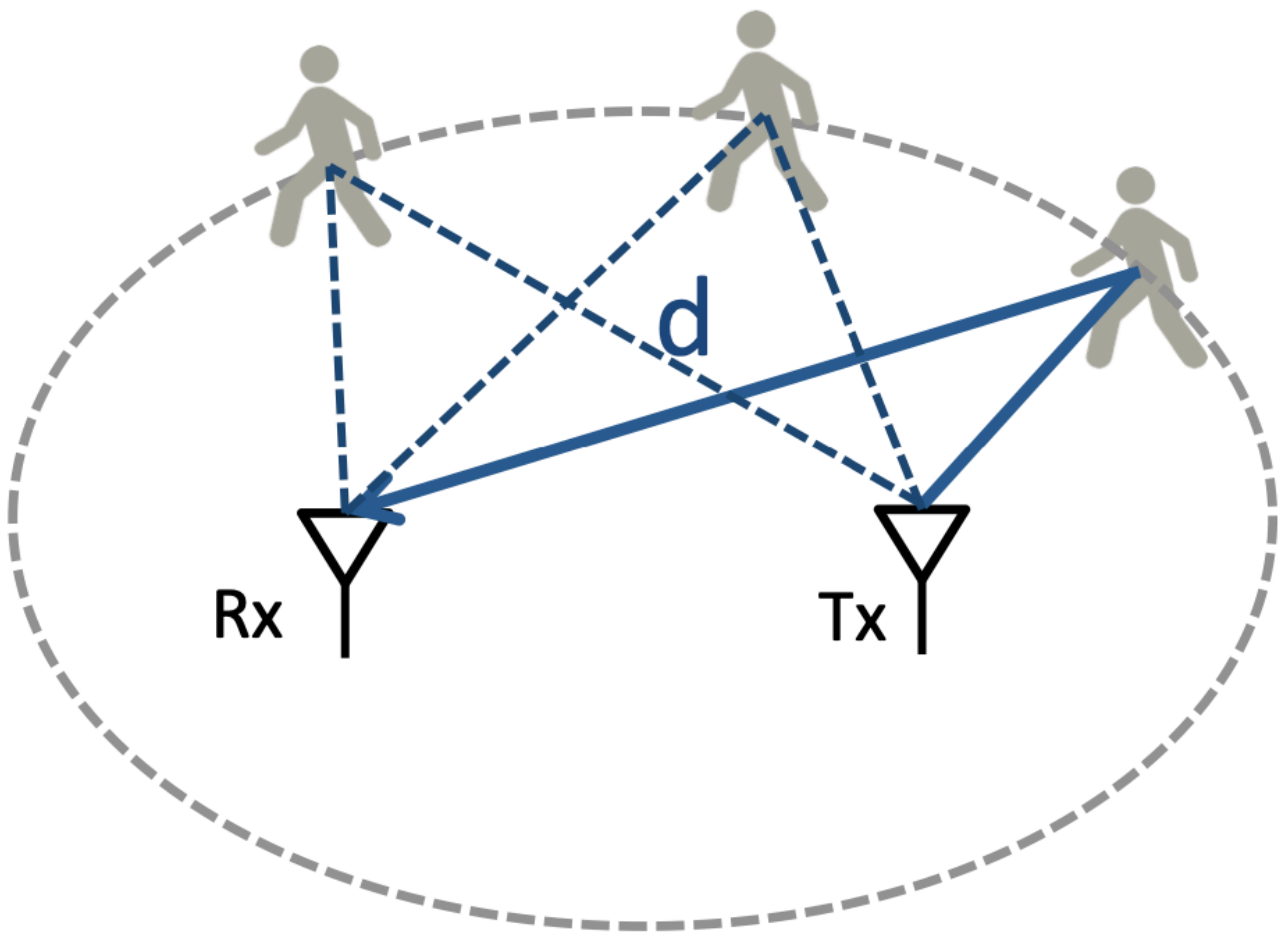


# Removing reflections by dynamic multi-path




**Key observation:** the direct reflection arrives before dynamic multi-path → the first peak represents the moving person

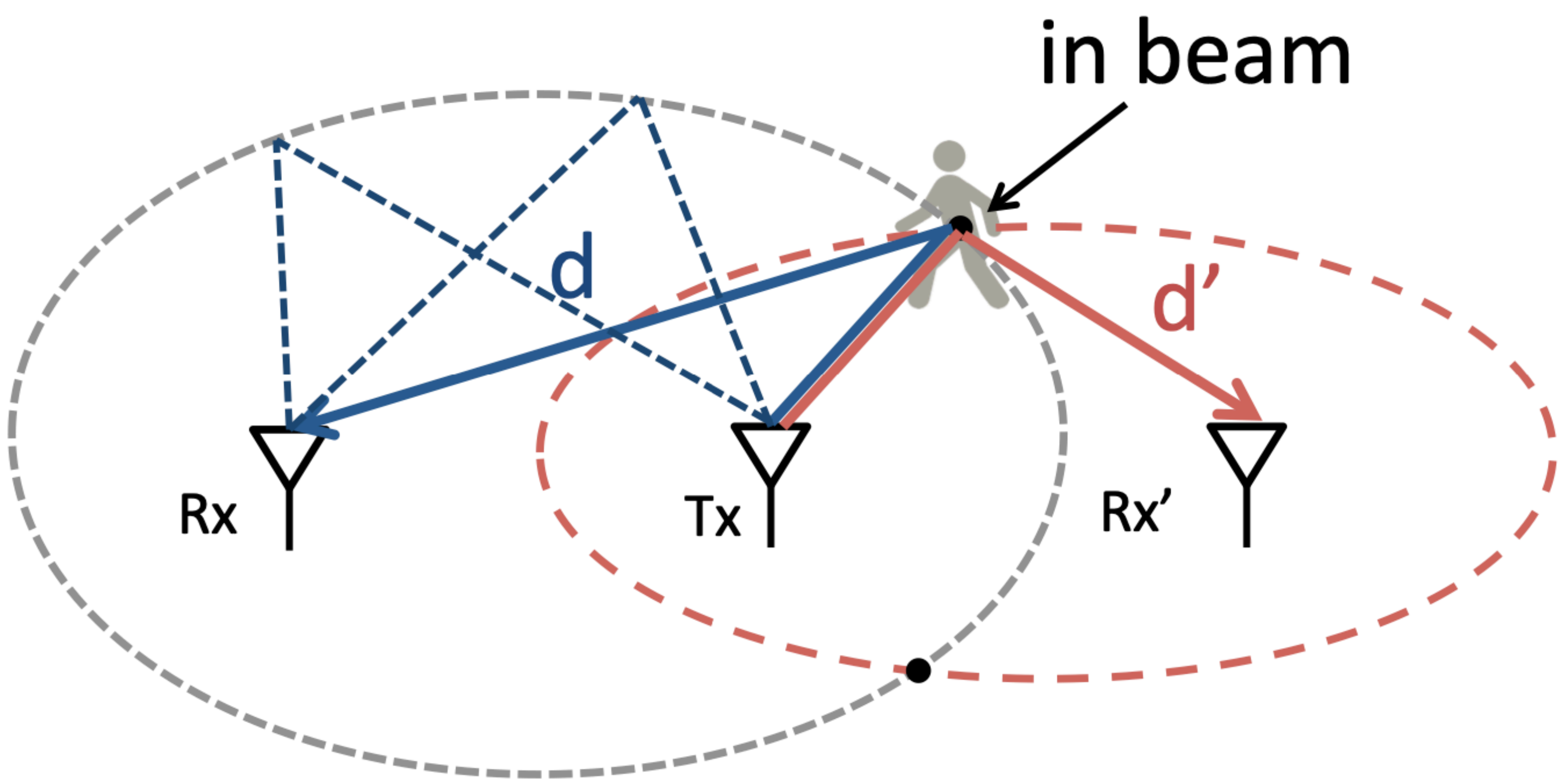
# From distance to localization




With the distance calculation, the person can be anywhere on an ellipse whose foci are (Tx, Rx) → One ellipse is not enough to localize!

# From distance to localization

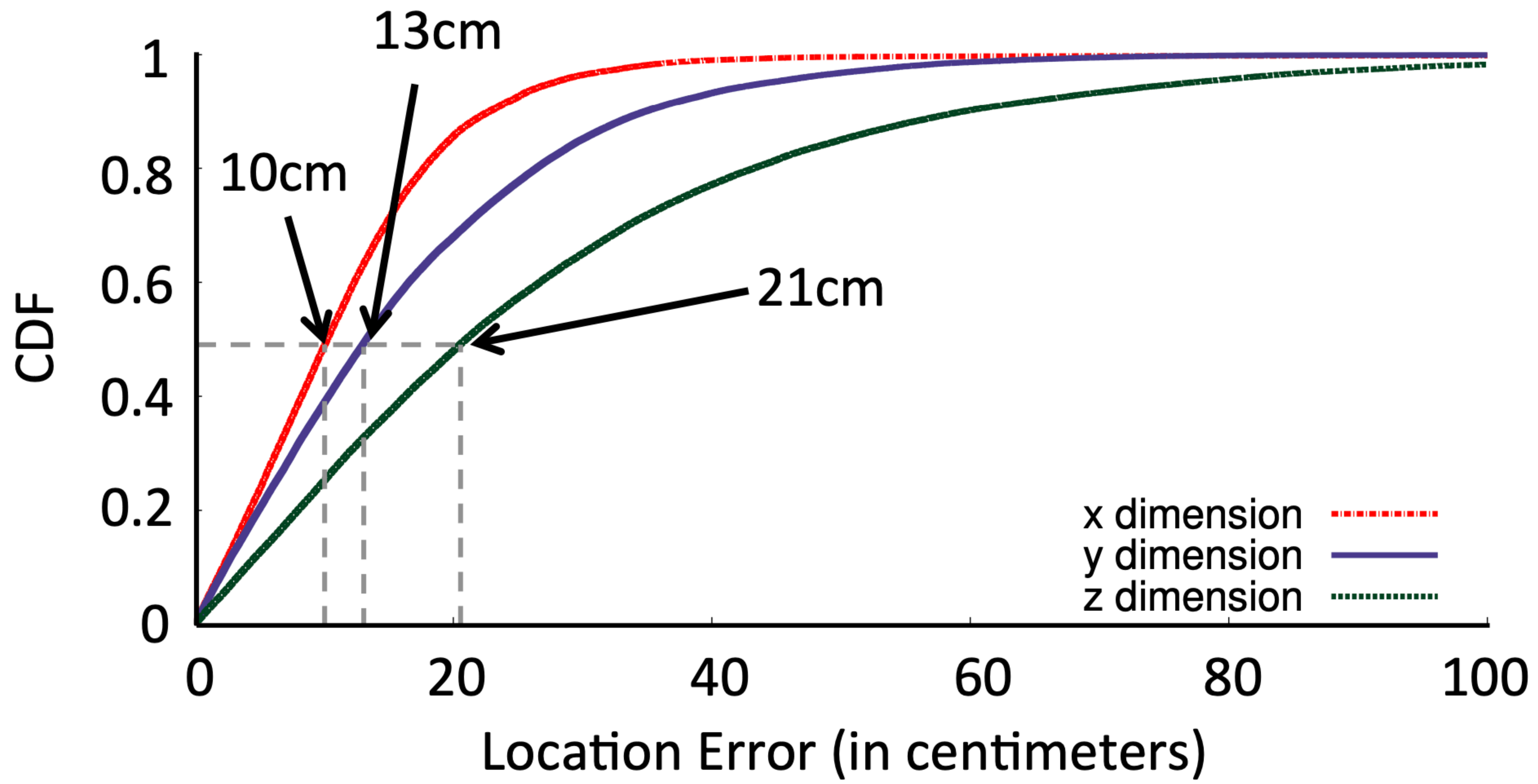
Use 2 Rx (i.e., two ellipse) and take the intersection of the ellipse to localize




There are two possible locations, but WiTrack uses directional antenna to locate the in-beam location

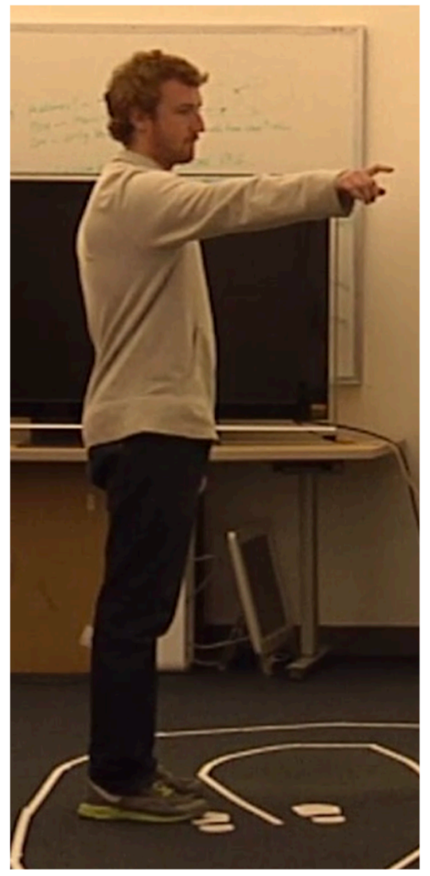
Can be extended to 3D localization if we use 3 Rx antennas and take the intersection of the 3 ellipsoids

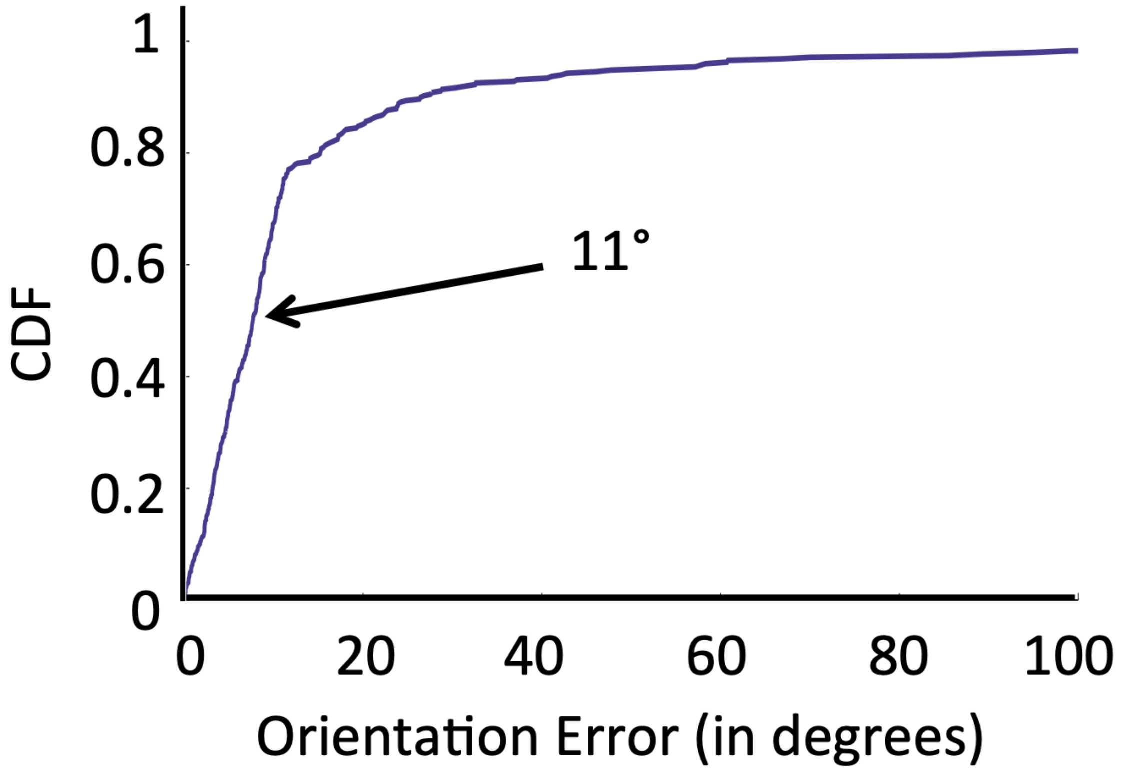
# Results



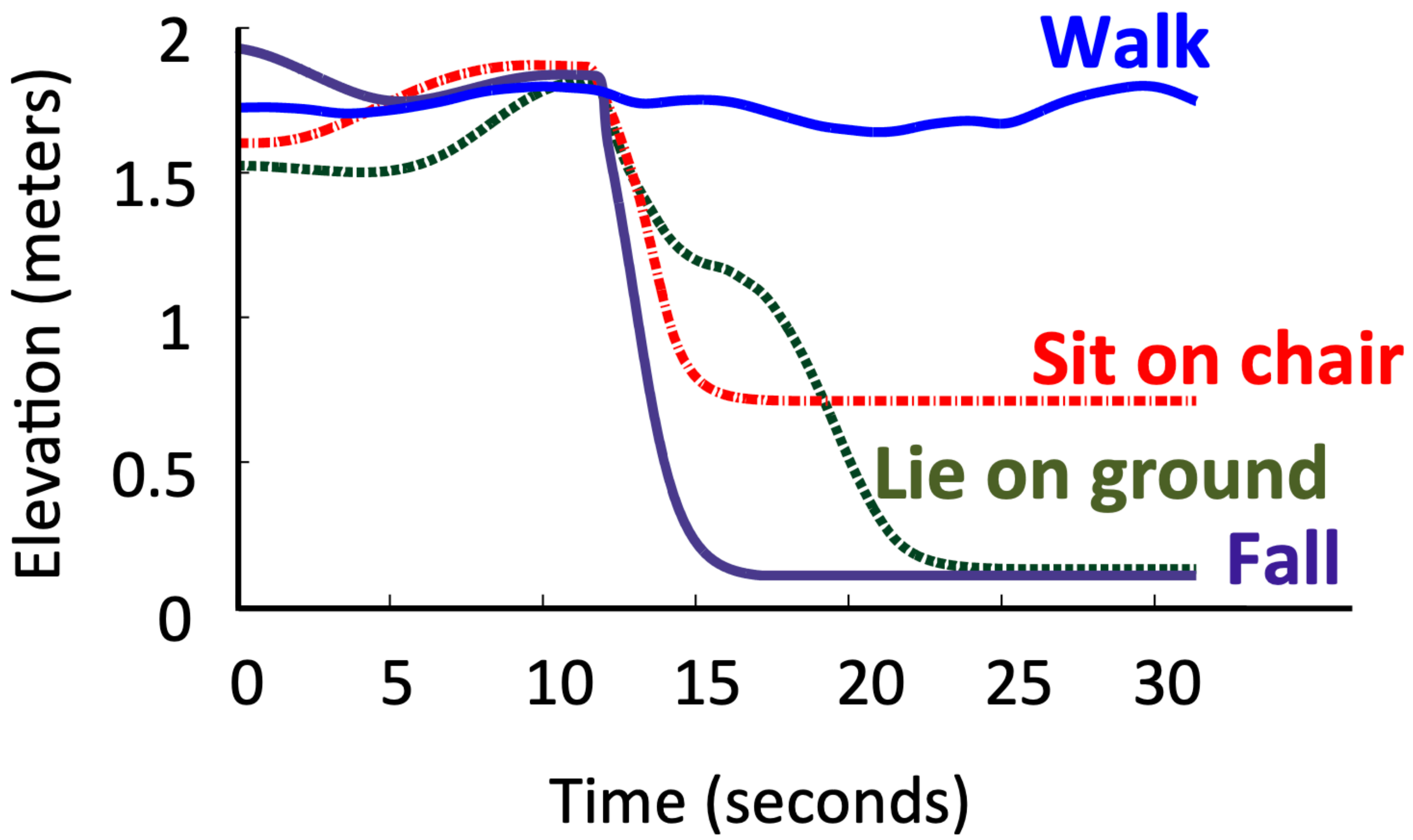

Centimeter-scale localization without requiring the user to carry a wireless device

# Other tasks






Person pointing to a random direction




Elderly monitoring to detect falls

# That is all!

# Course summary

**Introduction** (Internet history, course logistics and structure)

**Fundamentals**
Principles (end-to-end arguments, fate-sharing, packet vs. circuit switching), probabilistic data structures (Bloom filter, sketch)

**Forwarding and routing**
DNS, link layer (framing, error detection, MAC), Ethernet switching, ARP, network routing, router architectures

**Network transport**
TCP congestion control (slow-start, AIMD, variants, BBR), multi-path transport (design, congestion control), HTTP and QUIC

# Course summary

**Data center networking**
Architecture (fat-tree topology, addressing, routing), L2 vs L3, PortLand (idea, design)

**Data center transport**
Congestion (incast, flow control), MPTCP (idea, design), TIMELY (idea, design)

**Software defined networking**
Complexity in networking (lack of abstractions), SDN (idea, abstractions, OpenFlow), network virtualization (OVS, NVP)

**Programable forwarding**
Data plane programmability (why and how), P4, NetCache (idea, design)

# Course summary

**Video streaming**
Video encoding (compression, bitrate), streaming protocols (RTP, DASH standard), ABR algorithms (rate-based, buffer-based), VDN (idea, design)

**Video stream analytics**
Challenges (bandwidth, latency), AWStream (idea, design), DDS (idea, design)

**Networking for ML**
Networking challenges in deep learning, ByteScheduler (idea, design), Neurosurgeon (idea, design)

**ML for networking**
Reinforcement learning, Pensieve (idea, design), NeuroCuts (idea, design)

# THANK YOU

Final Q&A: Wednesday December 8, 11:00 - 12:45