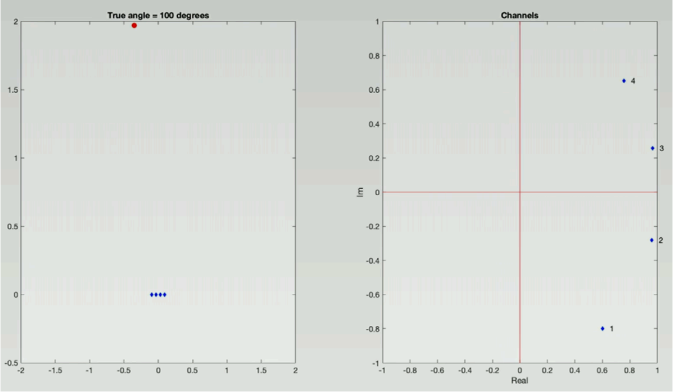
CS 598: Lecture 10

Instructor: Prof. Deepak Vasisht
Sep 27, 2024

Part 1: Multipath Profile Demo

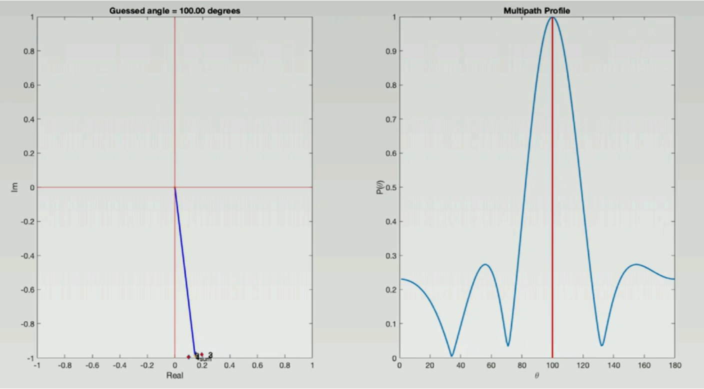
- Demo: Show the impact of different angles on different phase. Therefore, we can use the phase information to extract the right angle



- The phase difference between antennas is:

$$i = \frac{2\pi}{\lambda} ixcos(\theta) \tag{1}$$

Where x is the distance between antennas and theta is the angle



- If the guessed angle is correct, the phase differences are properly corrected, and the signals from the antennas will align, resulting in a large sum

  If the guess is wrong, the signals won't align, and the sum will be smaller
- By trying different angles and summing the corrected signals for each, the angle that results in the **largest sum** is the correct one. This process creates a "multipath profile" where the peak corresponds to the correct angle.

Part 2: Distance Measurements

Estimate the **AoA** (angle of arrival) by measuring phase on different antennas.

Estimate the **distance** by measuring phase on different frequencies/wavelengths

Phase and Distance

- The phase of a signal is dependent on both the signal's wavelength and the distance it has traveled. The formula given for phase is:

$$\angle h = \frac{2\pi}{\lambda} \cdot d \;\; mod \; 2\pi \tag{2}$$

where λ is the wavelength, d is the distance, h is the measured phase at a given frequency

- Wireless systems often operate across multiple frequency bands. For example, Wi-Fi operates in the 5 GHz band, but within this band, it can use different channels (e.g., 5.2 GHz, 5.3 GHz, etc.)
- By measuring the phase across these different frequency channels, a "multipath profile" can be created that helps estimate the true distance

Multipath Profile for Distance

- Just like you can create a multipath profile for angle (to see which angles have strong signals), you can create a **multipath profile for distance**, showing the signal strengths at different distances

$$h_i = a_i e^{-j\frac{2\pi}{\lambda_i}d'} \tag{3}$$

- This is the **channel response** at the i-th frequency, where hi is a complex number representing both amplitude ai and phase shift caused by the signal traveling a distance d'

$$P(d) = \sum_i h_i e^{j\frac{2\pi}{\lambda_i}d} \tag{4}$$

- This equation sums the channel responses from all different frequencies (indexed by i) to estimate a **distance profile** P(d). If the correct distance d is applied, the phases of the different frequencies will align, resulting in a peak in P(d), indicating the estimated distance between devices

Device localization

- By obtaining both the angle and distance between two devices, it's possible to localize a device using a single device with multiple antennas. The angles provide directional information, and the distance helps narrow down the exact location by "drawing circles" based on the distance. This combination allows for accurate device localization in a wireless system

Part 3: CFO Errors

Clock Frequency Offsets

- Each device may have a slightly different clock frequency, leading to **Clock Frequency Offset (CFO)** errors. In antenna arrays, multiple antennas are measuring the signal's channel response (H1, H2, etc.), but the phase at each antenna may differ not only due to the angle of arrival but also due to **clock frequency differences**
- The solution in antenna arrays is to **share the same clock** across all antennas (since they belong to the same device), which eliminates phase offsets due to clock differences between antennas

CFO in Distance Estimation

- When measuring distance between devices (e.g., an access point and a phone), the CFO error can affect the **phase measurements**
- The phase measurements at different frequency bands (e.g., sending a packet at frequency F1, measuring H1, then sending at F2 and measuring H2) can be corrupted by CFO errors because the clocks may drift over time, causing differences in phase accumulation

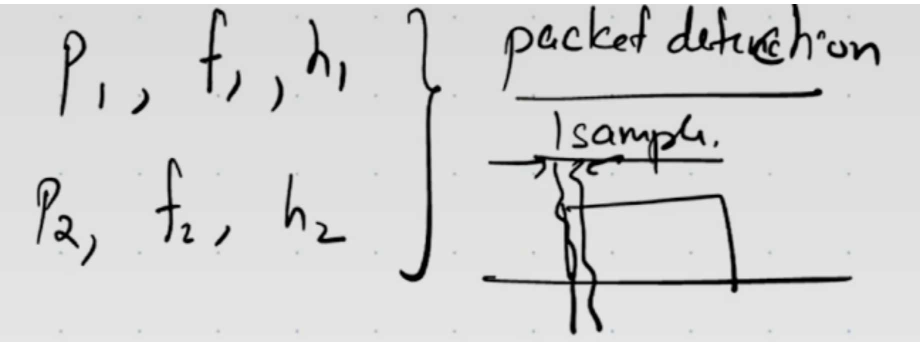
Chronos Paper's Solution

- The **Chronos paper** solves this problem by measuring phase both for a transmitted **packet** and its corresponding **ACK**
- The idea is that CFO affects the packet and the ACK **in opposite directions**. If the CFO for the packet is ΔF, the CFO for the ACK will be −ΔF, thus canceling out the CFO when both are considered together
- By averaging the phases of the packet and ACK, the CFO error can be eliminated, leaving only the true phase related to the distance between devices

> The CFO may change over time, which limits how long a packet can be. If the packet is too long, the CFO may drift even within the same transmission. However, Chronos assumes that during the short time interval between the packet and ACK, the CFO remains stable, allowing the phase correction method to work effectively

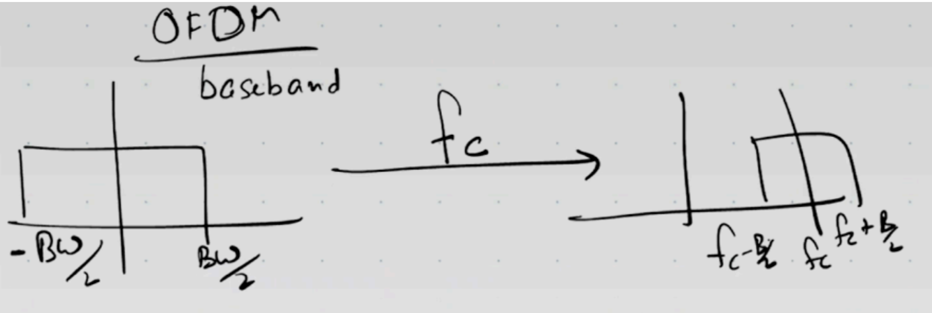
Part 4: Packet Detection Delay

Phase Measurement and Packet Detection



- If the **packet detection** process is off by one or two samples, it introduces **additional delay**, which results in a **phase shift**
- This delay affects phase measurements, especially when working across multiple frequency bands. Chronos needs to correct for these **phase errors** caused by the delay to ensure accurate distance or angle estimations

High-Level Idea of Chronos' Solution

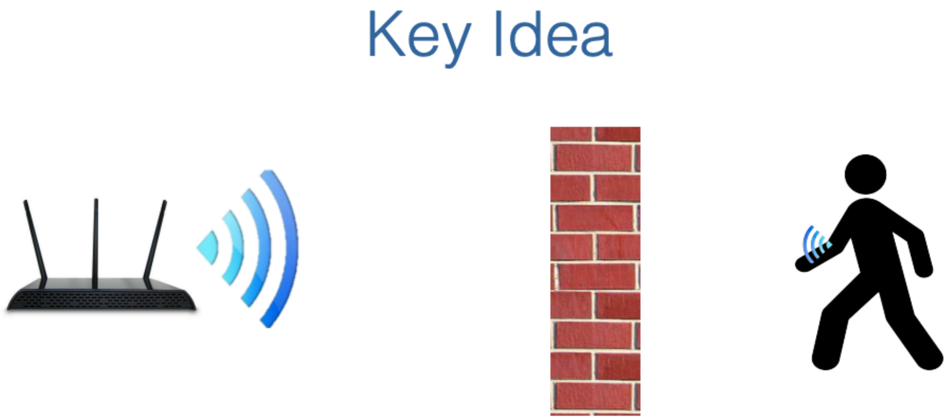


- In an **OFDM system**, signals are created in the baseband (from −B/2to B/2, where B is the bandwidth) and then upconverted to a higher **carrier frequency (Fc)** for transmission.
- At the receiver, the signal is **downconverted** by subtracting the carrier frequency, and then digital processing, including packet detection and CFO correction, is applied
- The phase at the **center frequency** (the middle of the bandwidth) remains unaffected by the **digital processing** that happens after downconversion. This is important because any errors in packet detection, extra delay, or randomness will not affect the phase at the center frequency.

  Chronos focuses on this **center frequency phase** for its computations, as it provides a more reliable measure of the true phase, free from the distortions introduced by the digital processing. Only looking at the phase at the center frequency helps Chronos handle CFO errors and avoid complications that arise from using subcarriers within the same band

Part 5: WiTrack

Key Idea



Key Idea

- **WiTrack** relies on analyzing reflections from human bodies using wireless signals. The wireless signals are transmitted, reflect off obstacles (such as walls, furniture, and human bodies), and are then received back. The reflections from human bodies are processed to infer their location and movement, even behind walls
- Difference from previous systems: In previous systems, the tracked person needed to carry an **active device** (like a phone) to measure signal characteristics. However, WiTrack does not require the person to carry any device—the system simply analyzes the reflections caused by their presence and movement

Key Challenges

Challenges

- Strong Wall Reflections: Reflections from walls are **10,000 times stronger** than those from behind the wall (where a person might be). This makes it challenging to isolate the weaker reflections from the person
- Tracking people from their reflections

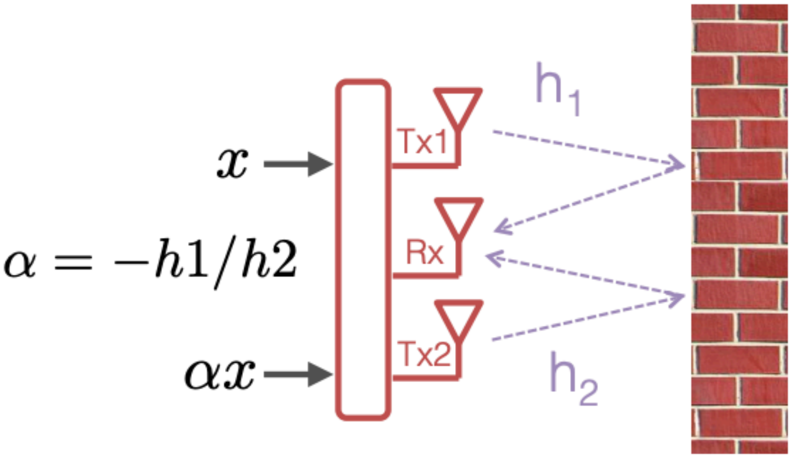
Eliminate the wall's reflection

Wall is static → disappears

People tend to move → detectable

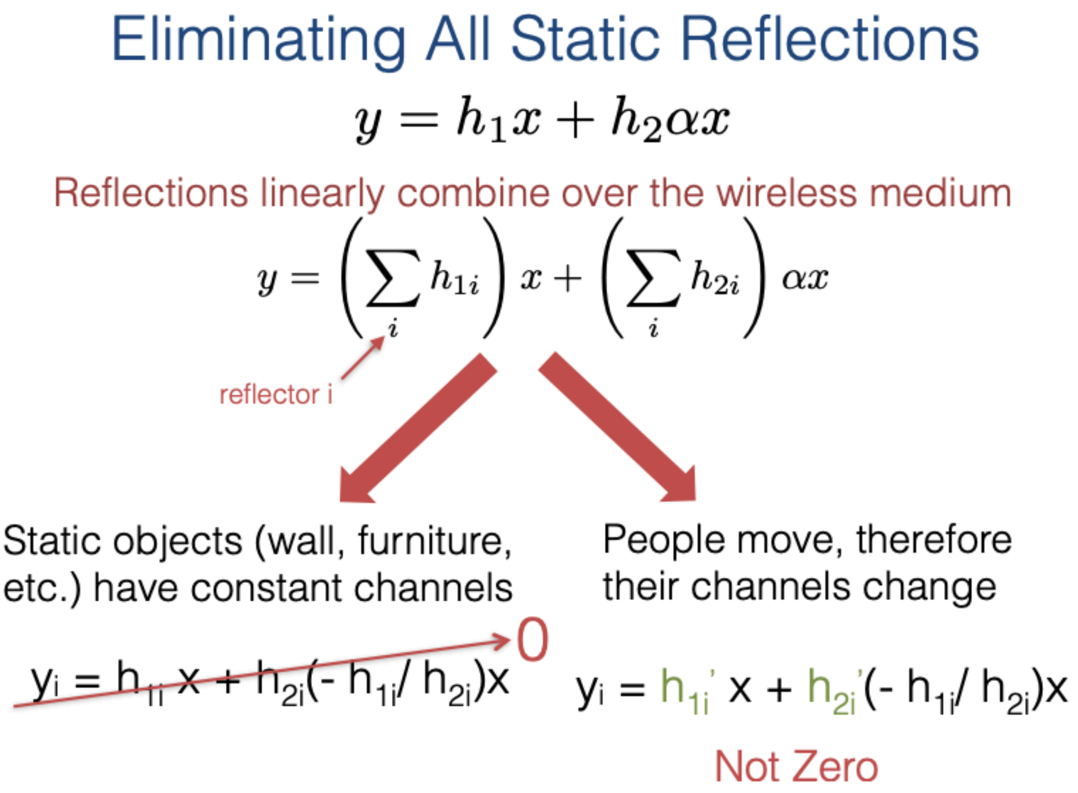
Transmit two waves that cancel each other when they reflect off static objects but not moving objects

Received signal: $\cancelto{0}{y = h_1 x + h_2 \alpha x}$



Two transmit antennas and one receive antenna, and set

$$\alpha = -h1/h2 \tag{5}$$



Eliminating All Static Reflections

$$y = h_1 x + h_2 \alpha x$$

Reflections linearly combine over the wireless medium

$$y = \left(\sum_i h_{1i}\right) x + \left(\sum_i h_{2i}\right) \alpha x$$

reflector i

Static objects (wall, furniture, etc.) have constant channels

$$\cancel{y_i = h_{1i} x + h_{2i}(- h_{1i}/ h_{2i})x} \rightarrow 0$$

People move, therefore their channels change

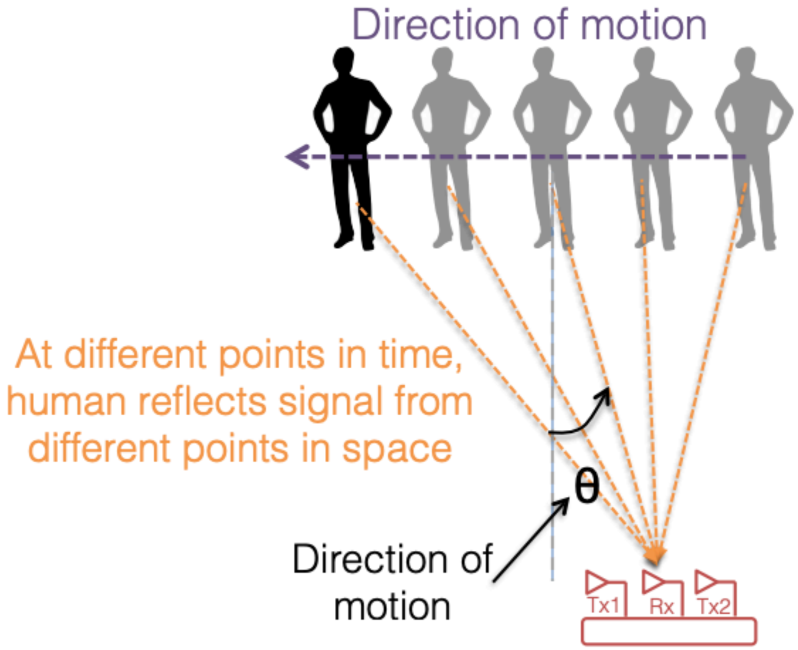
$$y_i = h_{1i}' x + h_{2i}'(- h_{1i}/ h_{2i})x$$

Not Zero

- **Static objects** (e.g., walls, furniture) have **constant channels** because they do not move. Their reflections can be represented as h1i and h2i for each static object i
- Since the values of h1' and h2' have changed due to movement, the reflections from moving objects do **not cancel out**, allowing the system to detect them

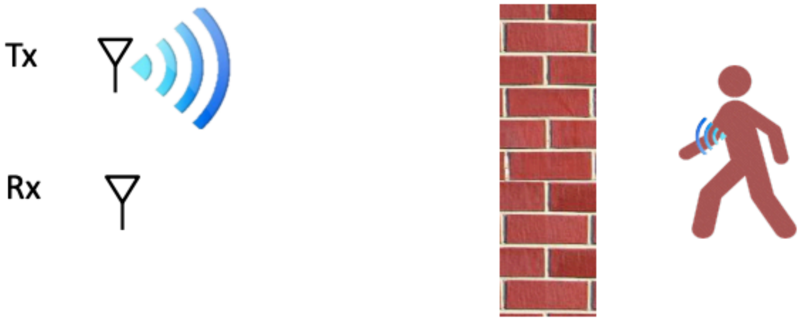
Track Using Reflections

Preliminary idea



- As a person moves, they reflect signals from **different points in space** at different points in time. Each position they take acts like a virtual antenna. Over time, the system captures reflections from different locations of the person, which effectively mimics the effect of having an **antenna array**. This virtual array created by human motion helps WiTrack gather multiple reflections over time, improving the accuracy of motion tracking
- Cons: Cannot track mutiple humans effectively
- That's why WiTrack comes. **WiTrack** tries to get the actual location of the human instead of just the direction.

Measure Distance

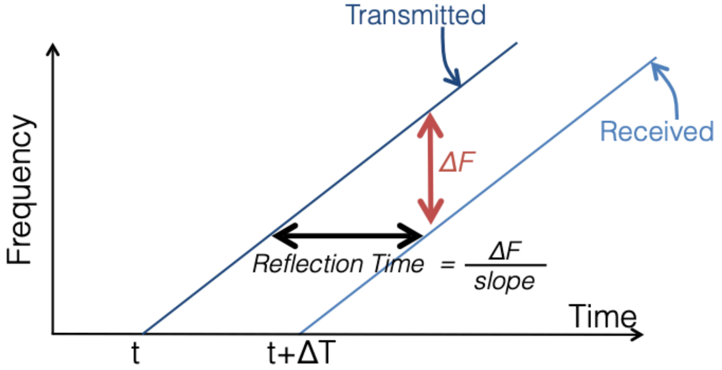


Distance = Reflection time x speed of light

- Calculate distances using time of flight
- How to measure reflection time?
  1. Transmit short pulse and listen for the echo

     But need to sample at very high rate. Multi-GHz samplers are expensive and generate high noise: not suitable for this application
  2. FMCW

FMCW: Measure time by measuring frequency



$$\text{Reflection Time} = \frac{\Delta F}{\text{slope}}$$

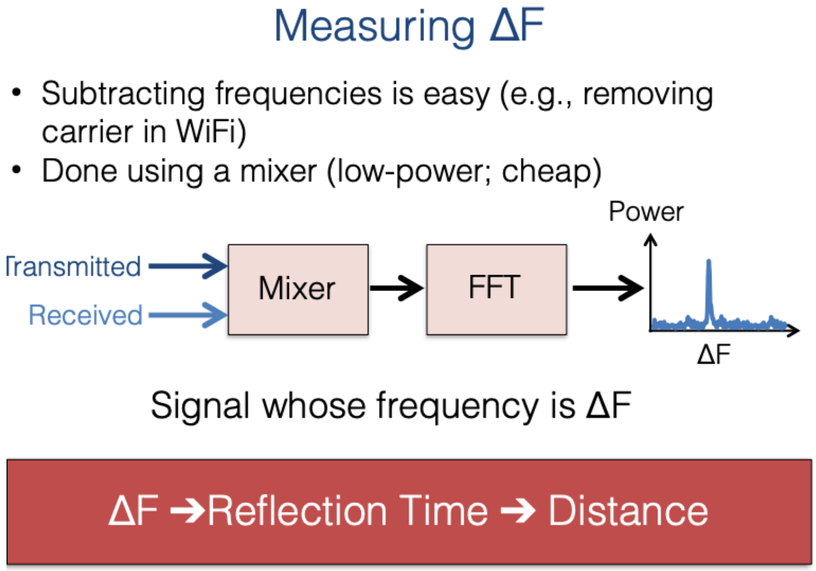
The signal is a **chirp**, where the frequency of the transmitted signal **increases linearly** over time. This means that at any given point in time, the transmitted signal is at a slightly different frequency compared to the reflected signal (because of the delay caused by the round trip)

The **frequency difference** between the transmitted and received signals is proportional to the time delay (ToF)

By measuring the **frequency difference** between the transmitted and received signals, and knowing the **slope** of the chirp (rate of change of frequency over time), WiTrack can calculate the **reflection time**

$$Refelction\ Time = \frac{\delta F}{slope} \tag{6}$$

This technique doesn't require precise time clocks, reducing the **cost** and **power consumption**



Measuring ΔF

- Subtracting frequencies is easy (e.g., removing carrier in WiFi)
- Done using a mixer (low-power; cheap)

Signal whose frequency is ΔF

ΔF ➔Reflection Time ➔ Distance

- Measuring ΔF: The system passes the **transmitted signal** and the **received signal** through a **mixer**, which outputs a signal whose frequency is ΔF

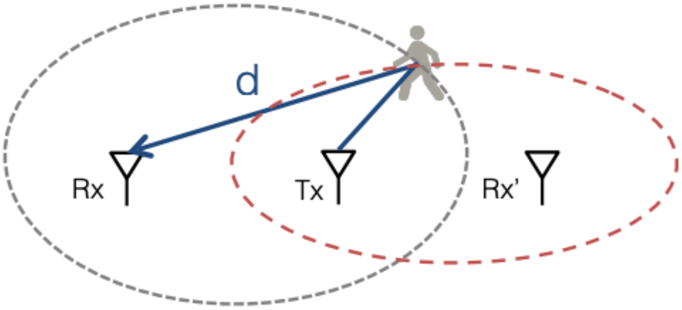
  The signal is then processed using a **Fast Fourier Transform (FFT)** to identify the frequency difference (ΔF) in the frequency domain. This is seen as a peak in the output graph (shown on the right of the slide)

  The frequency difference (ΔF) is proportional to the **reflection time (ToF)**

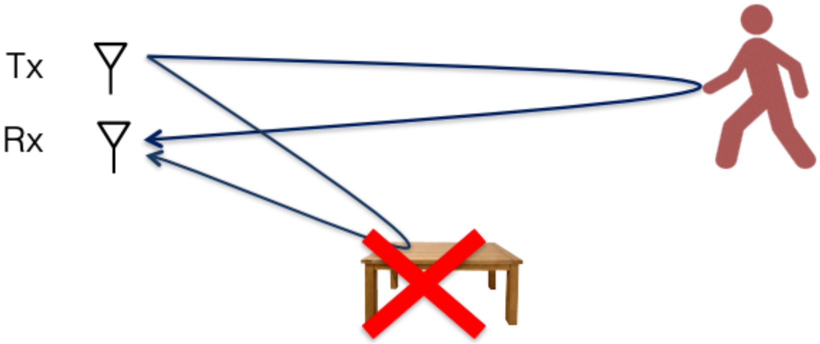
  Once ΔF is measured, it can be used to calculate the **reflection time** (ToF), which is the time it took for the signal to travel to the human body and back to the receiver

  By multiplying the reflection time by the **speed of light**, WiTrack can compute the **distance** between the person and the system

- Map Distance to Location: WiTrack uses the concept of **ellipses** to map distance to location. By measuring the round-trip distance (Tx to person to Rx), it knows the person is somewhere on the ellipse formed by Tx and Rx. Adding another receiver (Rx') allows the system to intersect multiple ellipses, which helps in determining the exact location of the person



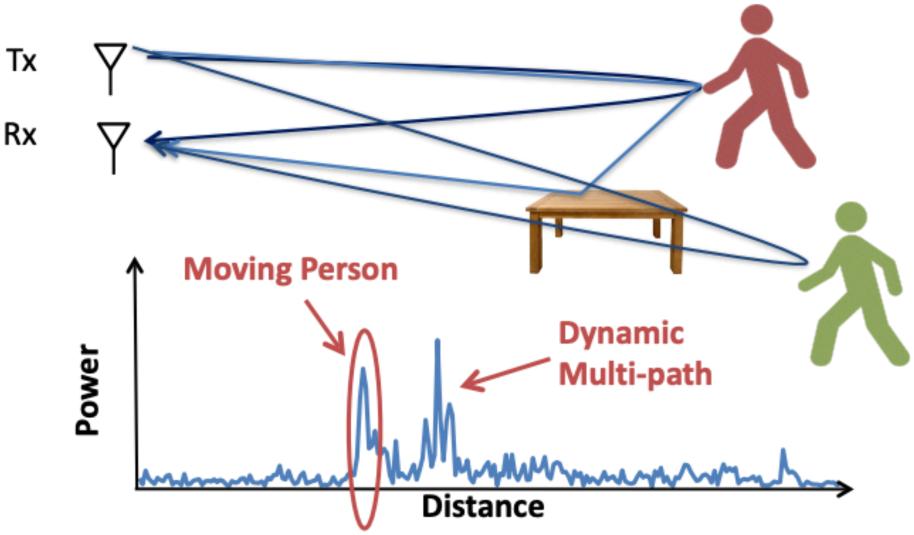
Multipath



- **Static Multipath**: When tracking a moving person, multipath reflections occur because the transmitted signal reflects off various objects (e.g., furniture) before reaching the receiver

  Direct furniture reflections can be eliminated by **subtracting consecutive measurements**, a process that helps remove reflections from **static objects** like tables

  Note: This approach requires the **user to move** for it to work effectively

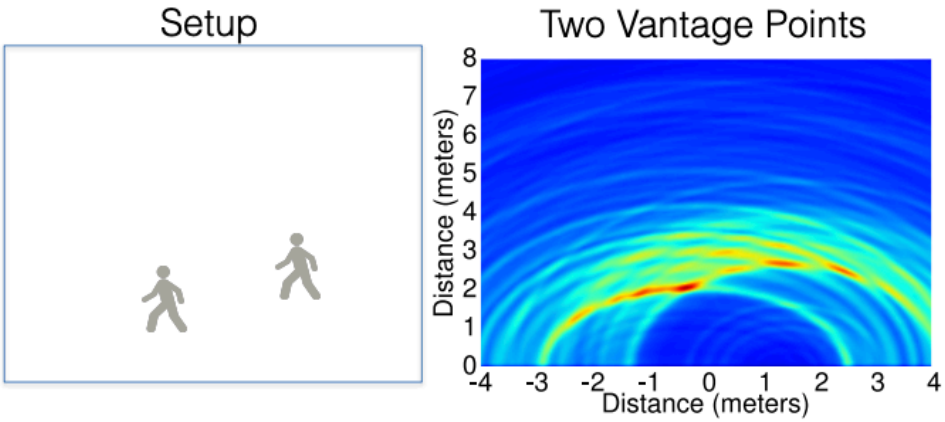


- Dynamic Multipath: The approach of subtracting static reflections works well for one moving user, but when **multiple people** are present, the situation becomes more complicated

  The **power-distance profile** (the graph at the bottom) shows different reflections over distance. The reflection from the **moving person** is a distinct peak, but **dynamic multipath** caused by other people or objects moving in the environment complicates the profile

- **Idea**: **combining multiple vantage points**

  In the right-hand graph (single vantage point), a **round-trip distance** is mapped to an **ellipse**, where the foci are the transmitter and receiver. However, using only one vantage point creates **confusion** in the data, making it difficult to differentiate between the users and multipath reflections



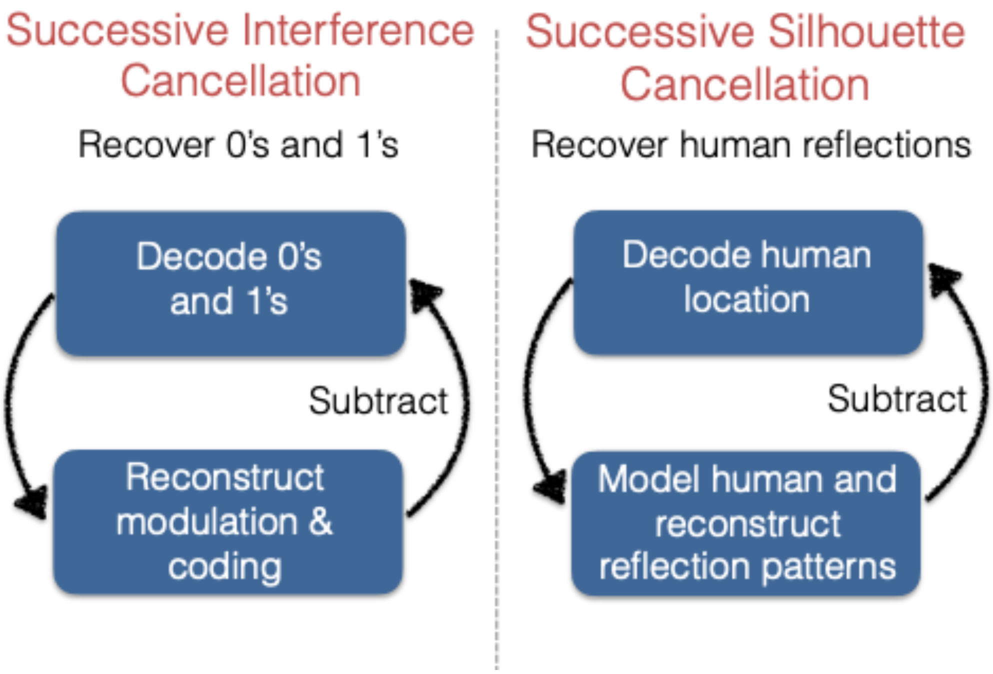
  When the system uses **two vantage points**, it can better disambiguate the direct paths of the two walking users by combining the data from both vantage points. This results in a clearer, more accurate representation of each person’s movement, as it helps to separate the actual reflections from the **multipath effects**

Multi-User Location

- **Successive Silhouette Cancellation (SSC)** is a new algorithm designed to localize multiple people in a scene by effectively handling the Near-Far Problem, where nearby individuals can mask distant ones.

  This algorithm is inspired by **Successive Interference Cancellation (SIC)**, which is commonly used in wireless communications to handle signal interference by iteratively decoding and canceling interfering transmissions. **Successive Silhouette Cancellation** iteratively decodes and removes human reflections from the total signal

- Successive Silhouette Cancellation Process:



Localize Static Users

Exploit Breathing Motion

- Breathing and walking occur at different time scales. The challenge here is that the system **cannot use the same subtraction window** to eliminate multi-path interference caused by walking and breathing. Since breathing happens much more slowly than walking, using the same subtraction method designed for faster motion like walking will not be effective for the slow, subtle motions caused by breathing
- Solution: **Use multi-resolution subtraction window** to eliminate multi-path while being able to localize both static and moving users.