CS60055: Ubiquitous Computing

Contactless Sensing

Part II: WiFi Sensing

Department of Computer Science and Engineering

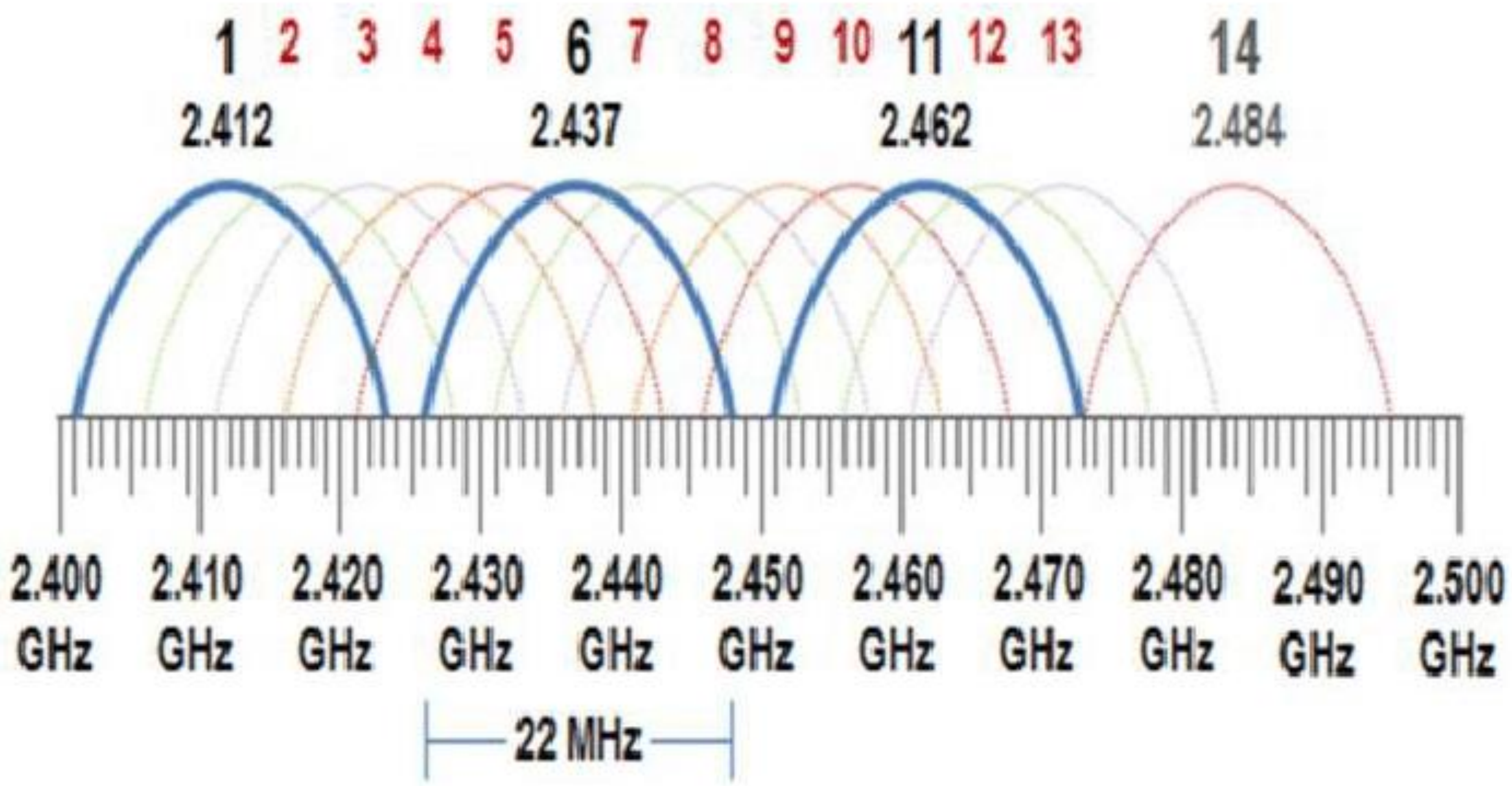
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Sandip Chakraborty
sandipc@cse.iitkgp.ac.in

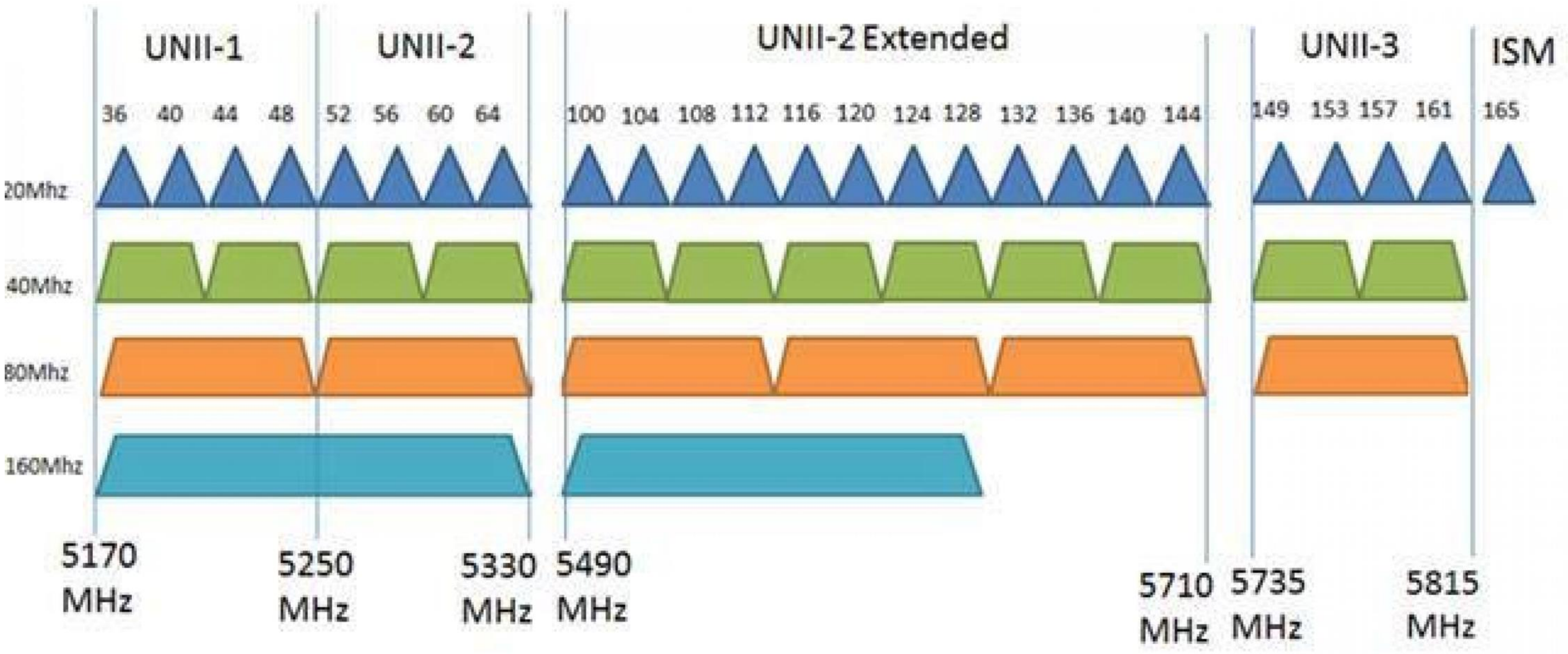
WiFi Sensing

Image Courtesy: Vijay Kumar Singh, IIIT Delhi

WiFi Channels – 2.4 GHz



WiFi Channels - 5GHz

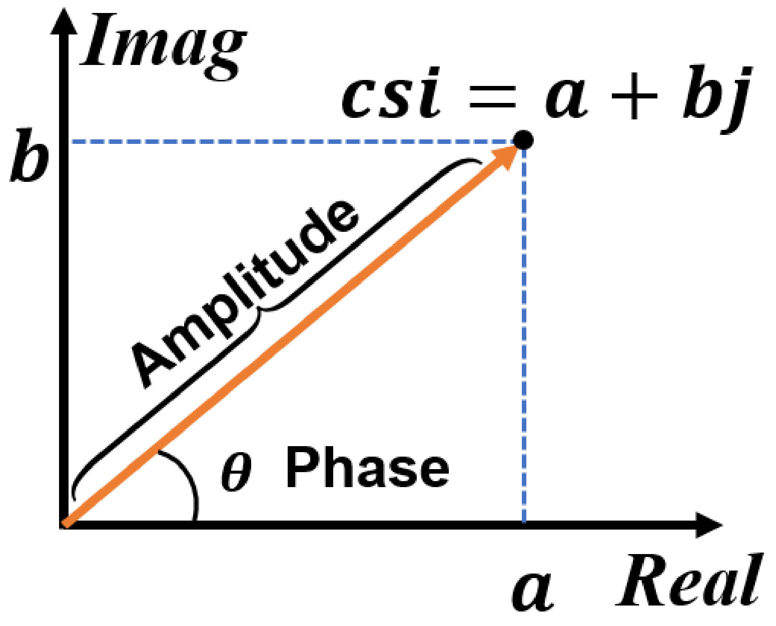


140 Channels, 24 non-overlapping

Channel State Information (CSI)

- Characterizes how wireless signal propagates from the transmitter to the receiver at certain carrier frequencies
  - CSI amplitude and phase are impacted by multipath effects (amplitude attenuation, phase shift)

- Represents the coefficient of a WiFi channel



- A packet transmitted with M transmitting antennas, N receiving antennas, 20MHz channel bandwitdh
  - CSI is a complex matrix of size MxNxZ where Z is the number of subcarriers in the target bandwidth

Channel State Information (CSI)

- Each CSI entry represents the channel frequency response (CFR):

$$H(f;t) = \sum_{n}^{N} a_n(t) e^{-j2\pi f \tau_n(t)}$$

  - $a_i(t)$ is the amplitude attenuation factor, $\tau_i(t)$ is the propagation delay, f is the carrier frequency, N is the total number of reflection path

- Wi-Fi NIC use several bits to represent the value of the real and imaginary numbers.
  - For example, Atheros Wi-Fi NIC uses 10 bits to give the value of the real and another 10 bits to describe the imaginary part of the number.

CSI Computation – Scattering Model

- Signal can be scattered through both the static and the dynamic objects in the environment
  - CSI observed by the receiver is added up with the contributions from the static and the dynamic scatterers
  - **You may think of each scatterer as a virtual TX**
  - Suitable for indoor scenarios, and is impacted by the dynamic movements within indoor – suitable for sensing movements

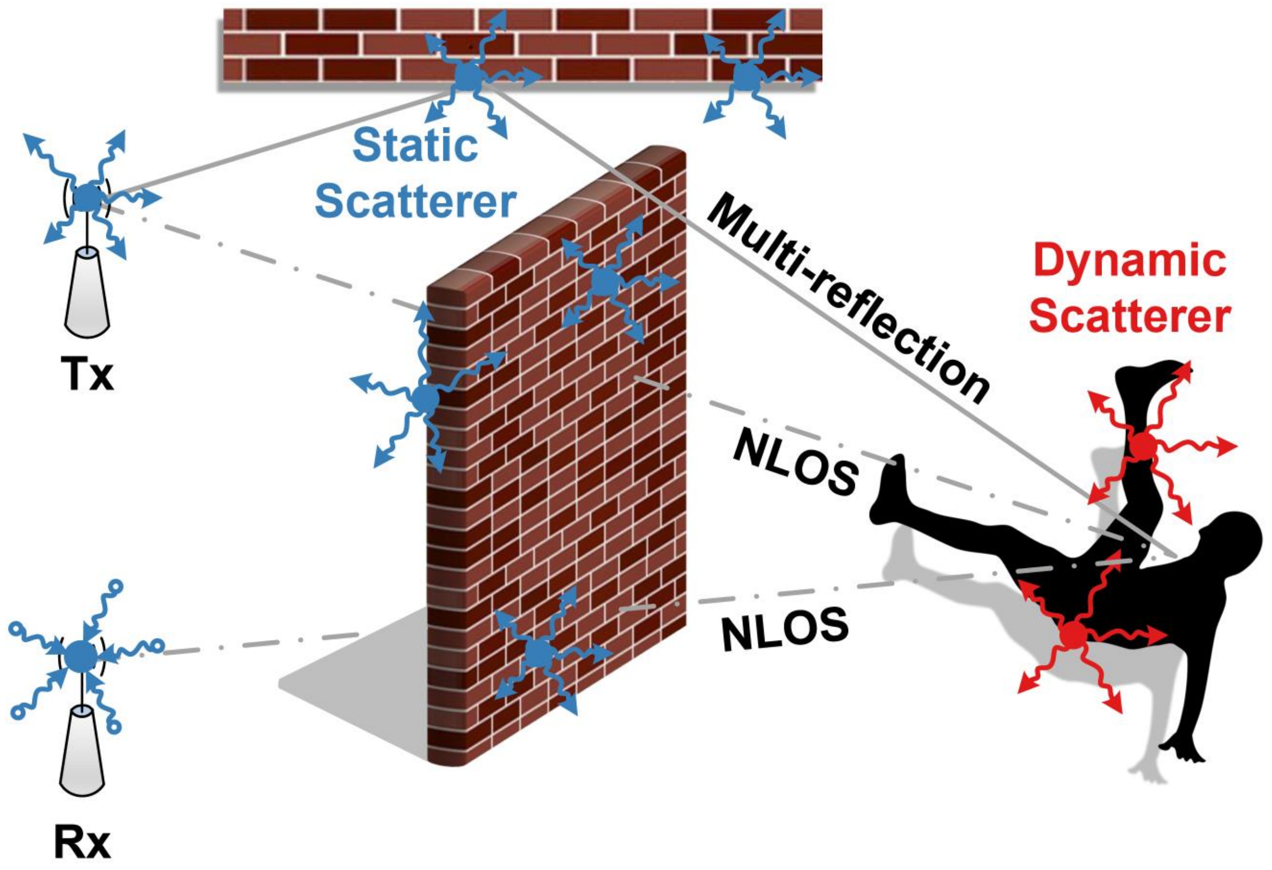


Image Source: https://tns.thss.tsinghua.edu.cn/wst/docs/pre/

WiFi Channel with MIMO and OFDM

- MIMO: Multiple Input Multiple Output
  - Use multiple TX and RX for spatial multiplexing
  - Can do a better estimation of multipath based on Line of Sight (LoS) and Non-line of Sight (nLoS) signal analysis

- Orthogonal Frequency Division Multiplexing (OFDM)
  - Rather than using a wideband, divide the wideband into multiple narrowbands, called the subcarriers
  - Helps in reducing attenuation loss and recover from channel errors
  - A specialized frequency division multiplexing
    - The subcarriers are orthogonal to each other

WiFi Tx and Rx

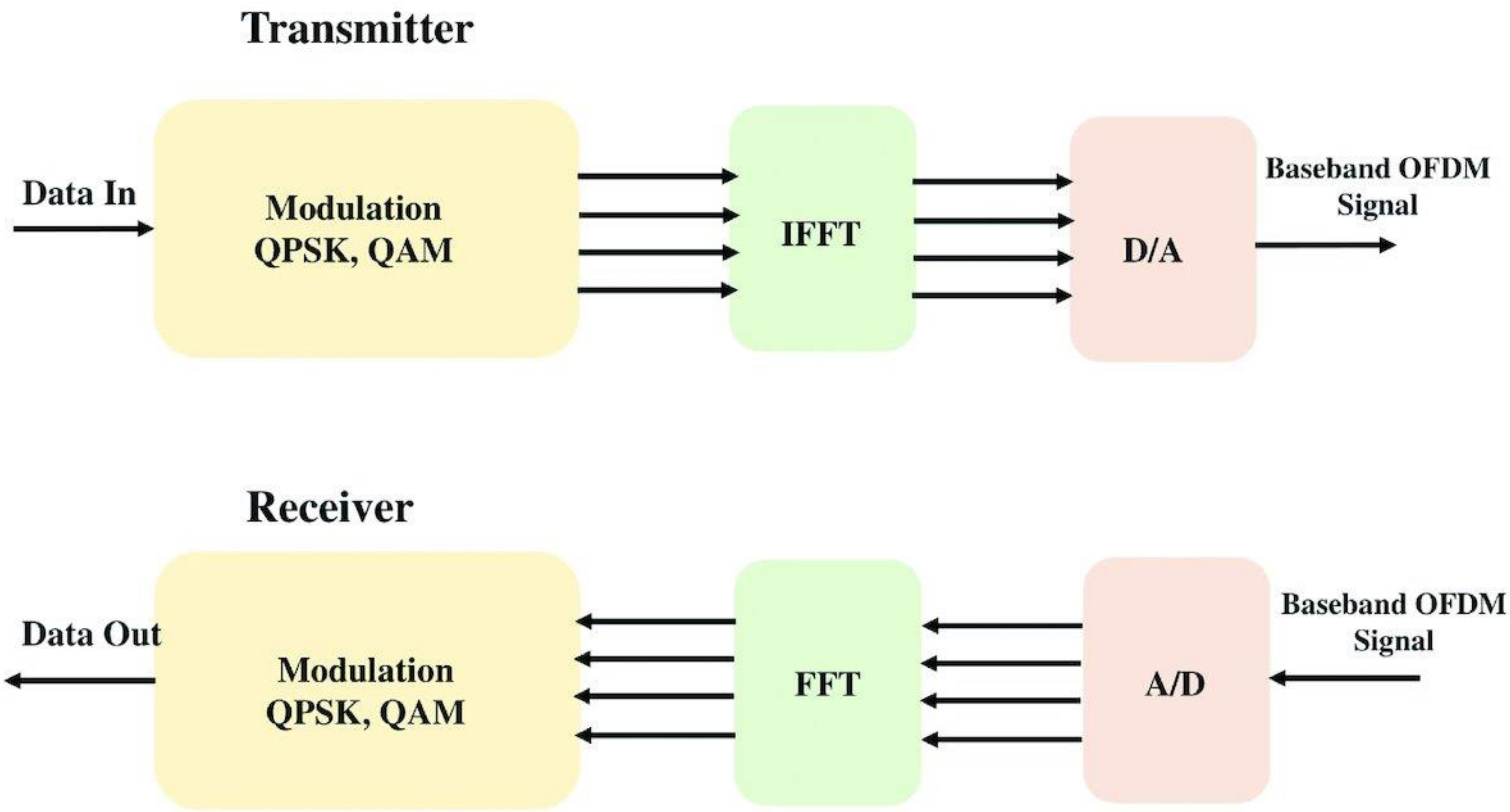
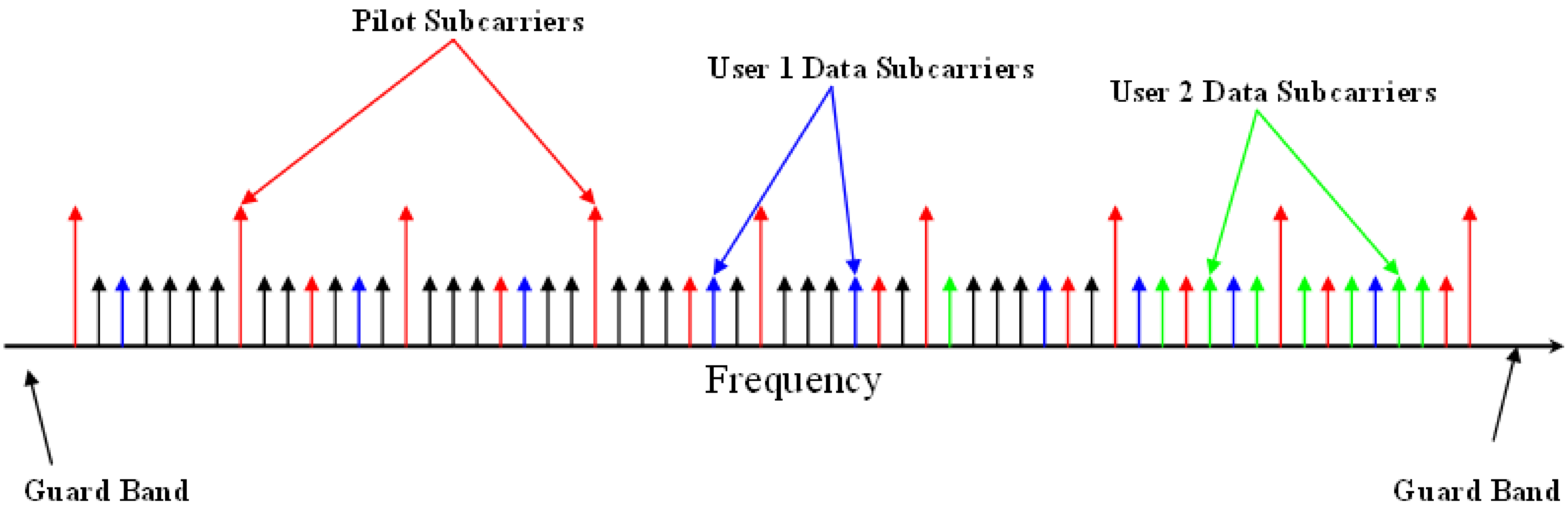


Image Source: https://doi.org/10.1093/jcde/qwaa048

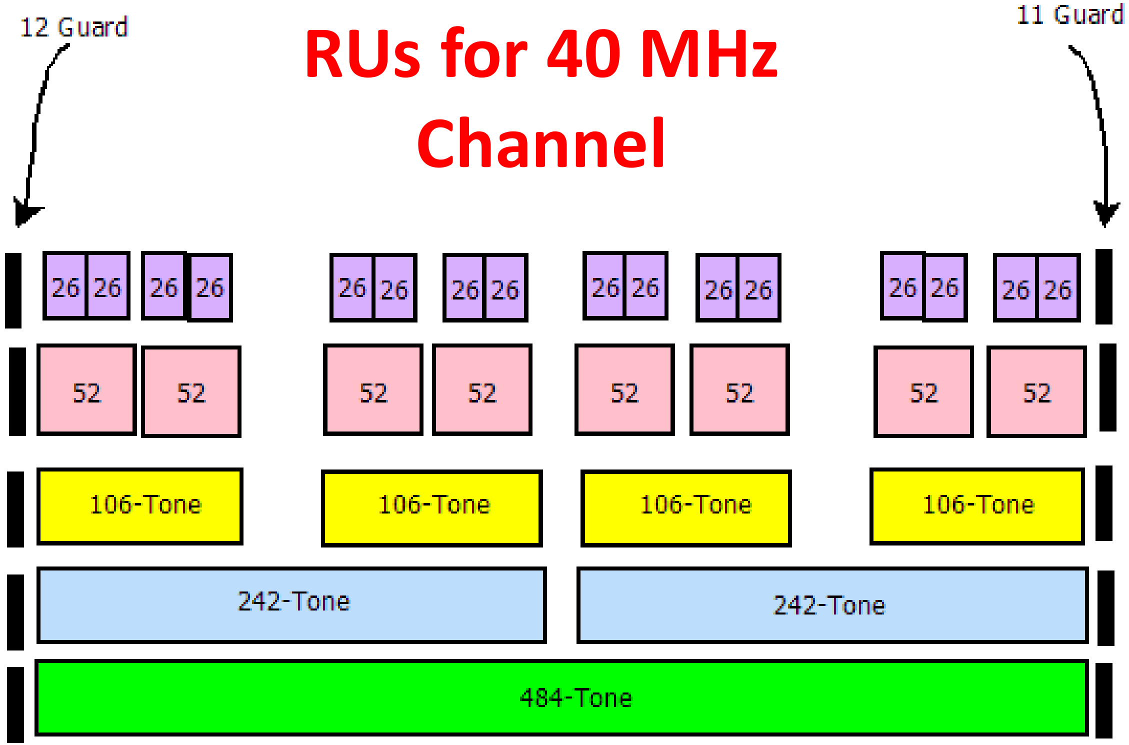
OFDMA in 802.11ax (WiFi 6)

- Use the subcarriers to transmit data from multiple users simultaneously



OFDMA in 802.11ax (WiFi 6)

- Use the subcarriers to transmit data from multiple users simultaneously
- The subcarriers are grouped together to form the resource units (RUs)
  - Different RUs can be assigned to different users, thus increasing the spectral efficiency significantly
- Three types of subcarriers are used
  - **Data subcarrier:** To transmit the actual data
  - **Pilot subcarrier:** Used for phase information and parameter tracking
  - **Unused:** DC, guard band, unallocated
- Different types of RUs are supported based on the bandwidth requirements



CSI Measurement for OFDM

- WiFi Tx transmits Long Training Symbols (LTFs)
  - Contains pre-defined symbols for each subcarrier, in the packet preamble
- When LTFs are received, Rx estimates the CSIs using the received signal and the original LTFs
- For each subcarrier, the WiFi channel is modeled as,

$$\boldsymbol{y} = H\boldsymbol{x} + \boldsymbol{n},$$

  - ***y*** is the received signal
  - ***x*** is the transmitted signal
  - *H* is the CSI matrix
  - ***n*** is the noise vector
- The estimated CSI is a 3-D matrix of complex values

CSI Measurement for OFDM

- In real world, the measured CSI is impacted by several factors
  - Multi-path channels
  - Transmit/receive processing
  - Hardware/software errors
- The measured baseband-to-baseband CSI can be represented as,

$$H_{i,j,k} = \underbrace{\left(\sum_{n}^{N} a_n e^{-j2\pi d_{i,j,n} f_k / c} \right)}_{\text{Multi-Path Channel}} \underbrace{e^{-j2\pi \tau_i f_k}}_{\substack{\text{Cyclic Shift} \\ \text{Diversity}}} \underbrace{e^{-j2\pi \rho f_k}}_{\substack{\text{Sampling} \\ \text{Time Offset}}} \underbrace{e^{-j2\pi \eta (f'_k / f_k - 1) f_k}}_{\substack{\text{Sampling} \\ \text{Frequency Offset}}} \underbrace{q_{i,j} e^{-j2\pi \zeta_{i,j}}}_{\text{Beamforming}},$$

  - $d_{i,j,n}$ is the path length from the i^{th} transmit antenna to the j^{th} receive antenna of the n^{th} path
  - f_k is the carrier frequency, ρ is the sampling time offset, η is the sampling frequency offset, $q_{i,j}$ and $\zeta_{i,j}$ are the amplitude attenuation and phase shift of the beamforming

CSI Measurement for OFDM

- In real world, the measured CSI is impacted by several factors
  - Multi-path channels
  - Transmit/receive processing
  - Hardware/software errors
- The measured baseband-to-baseband CSI can be represented as,

$$H_{i,j,k} = \underbrace{\left(\sum_{n}^{N} a_n e^{-j2\pi d_{i,j,n} f_k / c}\right)}_{\text{Multi-Path Channel}} \underbrace{e^{-j2\pi \tau_i f_k}}_{\substack{\text{Cyclic Shift}\\ \text{Diversity}}} \underbrace{e^{-j2\pi \rho f_k}}_{\substack{\text{Sampling}\\ \text{Time Offset}}} \underbrace{e^{-j2\pi \eta (f'_k / f_k - 1) f_k}}_{\substack{\text{Sampling}\\ \text{Frequency Offset}}} \underbrace{q_{i,j} e^{-j2\pi \zeta_{i,j}}}_{\text{Beamforming}},$$

  - $d_{i,j,n}$ is the path length from the
    n[th] path
  - f_k is the carrier frequency, ρ is
    offset, $q_{i,j}$ and $\zeta_{i,j}$ are the ampli

Signal Processing techniques are needed to remove the impact of CSD, STO, SFO, and beamforming

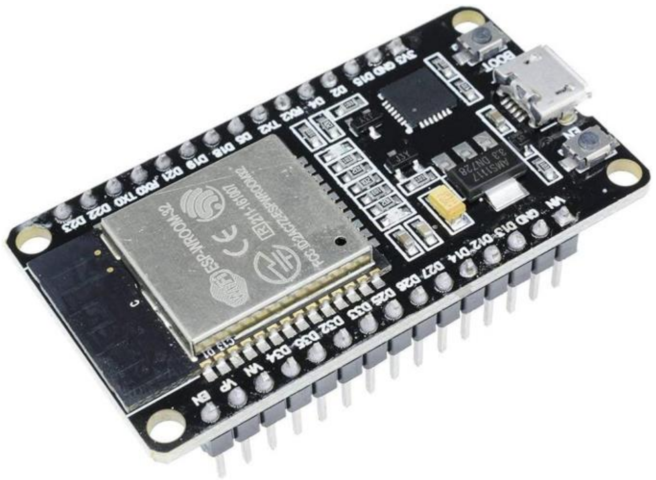
Obtaining CSI

- Not all COTS WiFi hardware allows exporting the CSI data; available for few cases --
  - 802.11 CSI tools (https://dhalperi.github.io/linux-80211n-csitool/): Uses Intel 5300
  - Atheros CSI tools (https://wands.sg/research/wifi/AtherosCSI/) - Qualcomm Atheros
  - ESP32 CSI toolkit (https://stevenmhernandez.github.io/ESP32-CSI-Tool/) - ESP32
  - Intel 5300 and Atheros provide much stable and unnoisy CSI values; however, works for restricted settings (802.11n)
  - In contrast, ESP32 is a cheap and easily available device; the data can be little noisy but works well for practical applications with suitable data pre-processing techniques applied

Visualizing CSI (ESP32 as a Use Case)

- Supports three CFR types
  - **Legacy long training field (LLTF)**: Uses the long training symbol for legacy wifi devices like 802.11a
  - **High-throughput LTF (HT-LTF)**: For high-throughput WiFi like 802.11n
  - **Space-time block code HT-LTF (STBC-HT-LTF)**: For MIMO settings

- You need to configure the CFR types and the mode of operations:
  - `active_sta`: Connect to some AP and send packet requests
  - `active_ap`: Works like an AP where other devices can connect
  - `passive`: Passively listens CSI frames on a given channel



Understanding the CSI Data – 20 MHz Band

- The 20 MHz band has 64 subcarriers
  - 48 subcarriers carry data (known as data subcarriers).
  - 4 subcarriers are pilot subcarriers (used for synchronization and channel estimation).
  - 12 subcarriers are either guard bands or unused.

- In ESP32, the real and imaginary components of CSI are typically represented as 8-bit signed integers (1 byte each)

- ESP32 captures CSI data for all 64 subcarriers, even though not all subcarriers are used for data transmission (some are guard bands and pilots)
  - So, you'll see some zeros in the CSI readings

Visualizing CSI (ESP32 as a Use Case)

- Each CFR of a subcarrier registers as two bytes of signed characters, the first part being the imaginary, and the second part is the real value
- CSI Information from different training fields:

Channel	Secondary channel	None			Below					Above				
Packet information	Signal mode	Non-HT	HT		Non-HT	HT				Non-HT	HT			
	Channel bandwidth	20 MHz	20 MHz		20 MHz	20 MHz		40 MHz		20 MHz	20 MHz	40 MHz		
	STBC	Non-STBC	Non-STBC	STBC	Non-STBC	Non-STBC	STBC	Non-STBC	STBC	Non-STBC	Non-STBC	STBC	Non-STBC	STBC
Subcarrier index	LLTF	0~31, −32~−1	0~31, −32~−1	0~31, −32~−1	0~63	0~63	0~63	0~63	0~63	−64~−1	−64~−1	−64~−1	−64~−1	−64~−1
	HT-LTF		0~31, −32~−1	0~31, −32~−1		0~63	0~62	0~63, −64~−1	0~60, −60~−1		−64~−1	−62~−1	0~63, −64~−1	0~60, −60~−1
	STBC- HT-LTF			0~31, −32~−1			0~62		0~60, −60~−1			−62~−1		0~63, −64~−1
Total bytes		128	256	384	128	256	380	384	612	128	256	376	384	612

Visualizing CSI (ESP32 as a Use Case)

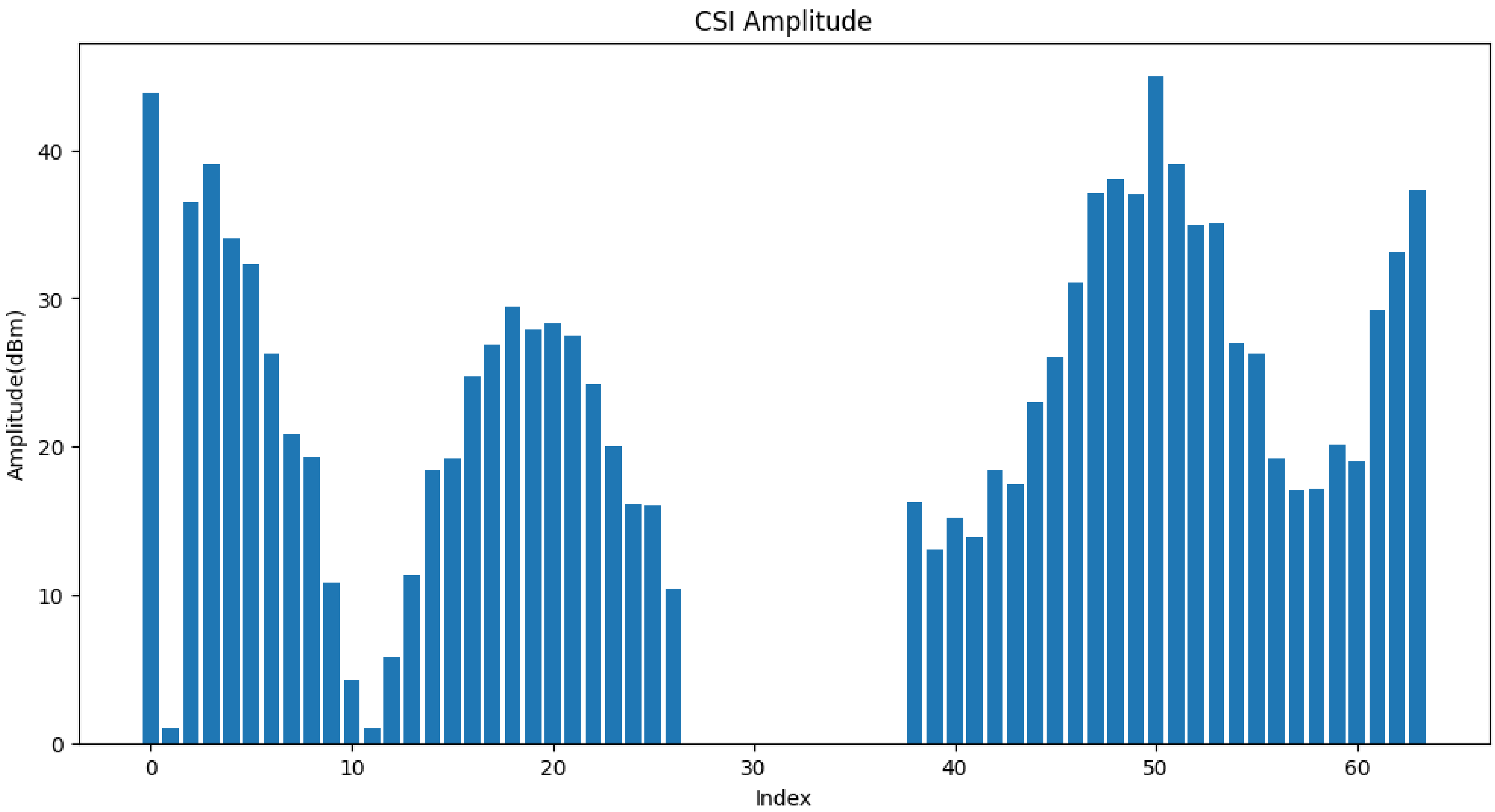

```
type,role,mac,rssi,rate,sig_mode,mcs,bandwidth,smoothing,not_sounding,aggregation,stbc,fec_coding,sgi,noise_floor,ampdu_cnt,channel,
secondary_channel,local_timestamp,ant,sig_len,rx_state,real_time_set,real_timestamp,len,CSI_DATA
CSI_DATA,PASSIVE,A0:46:5A:3F:05:93,-70,10,0,0,0,0,0,0,0,0,0,-96,0,6,0,26426,0,30,0,0,0.162907,128,[30 -32 1 0 21 11 21 11 21 12 22
10 23 12 22 12 21 10 22 10 23 10 25 11 23 10 20 10 22 11 21 11 21 12 22 13 19 13 18 13 19 14 19 14 18 14 17 14 16 14 16 14 12 16 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 13 9 13 7 13 8 14 7 13 7 13 6 13 7 14 7 13 7 14 7 14 8 14 7 13 7 14 9 13 9 14 9 12 10 11
11 12 13 14 13 14 15 12 15 13 18 13 19 13 17 12 ]
CSI_DATA,PASSIVE,FE:C4:E6:FA:26:44,-78,10,0,0,0,0,0,0,0,0,0,-96,0,6,0,79086,0,30,0,0,0.214742,128,[30 -32 1 0 36 6 39 2 34 0 32 -4
25 -8 20 -6 18 -7 9 -6 3 -3 0 1 -5 3 -8 8 -13 13 -9 17 -13 21 -12 24 -9 28 -7 27 -4 28 -5 27 -3 24 -1 20 2 16 1 16 3 10 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -16 3 -13 -1 -14 -6 -12 -7 -9 -16 -7 -16 -1 -23 -1 -26 2 -31 3 -37 -1 -38 2 -37 1 -45 -1 -39 0 -35 1
-35 1 -27 4 -26 3 -19 11 -13 14 -10 20 -2 19 0 29 4 33 3 37 5 ]
CSI_DATA,PASSIVE,00:00:00:00:00:00,-66,12,0,0,0,0,0,0,0,0,0,-96,0,6,0,80339,0,38,0,0,0.220421,128,[38 96 2 0 -15 32 -14 31 -13 29
-13 28 -11 26 -9 25 -9 25 -8 23 -6 22 -5 22 -4 21 -5 19 -4 18 -2 18 -2 17 -1 15 -1 16 -2 15 -1 14 -1 14 -1 12 -1 11 -1 11 -2 11 -2
10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 11 -1 11 -1 12 -2 13 -3 15 -3 17 -4 17 -6 20 -6 22 -7 22 -8 24 -9 25 -10 26 -11 28
-11 29 -12 30 -13 31 -11 33 -12 33 -12 33 -13 34 -14 35 -14 37 -15 36 -15 35 -14 36 ]
CSI_DATA,PASSIVE,BA:6C:98:41:93:9E,-72,11,1,7,0,0,1,1,0,0,1,-96,2,6,0,81219,0,106,0,0,0.226245,256,[106 -96 6 0 18 15 20 13 21 16 23
14 22 14 22 13 21 13 23 14 21 13 22 12 23 14 22 12 21 16 20 16 20 15 19 15 17 16 16 15 16 16 14 15 14 15 12 13 12 13 11 14 11 14 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 15 12 13 15 11 15 14 17 12 15 13 17 11 17 12 17 10 20 11 19 10 21 10 20 11 21 10 21 11 23 8
20 10 19 10 18 10 20 12 18 13 19 13 18 12 17 15 15 15 17 20 17 0 0 35 28 41 25 36 34 42 31 39 24 42 25 44 27 37 25 46 26 44 30 48 30
41 31 41 29 40 28 40 27 32 31 31 32 33 30 33 34 30 35 25 31 30 26 28 27 18 29 23 29 20 33 17 34 7 27 0 0 0 0 0 0 0 0 0 0 0 0 0 0 26
27 21 28 19 35 21 32 20 33 20 32 16 35 21 37 17 37 12 42 17 44 13 44 11 42 18 47 12 44 17 43 15 48 15 40 15 43 15 44 12 39 14 37 22
36 19 37 26 40 23 32 31 30 30 37 ]
```

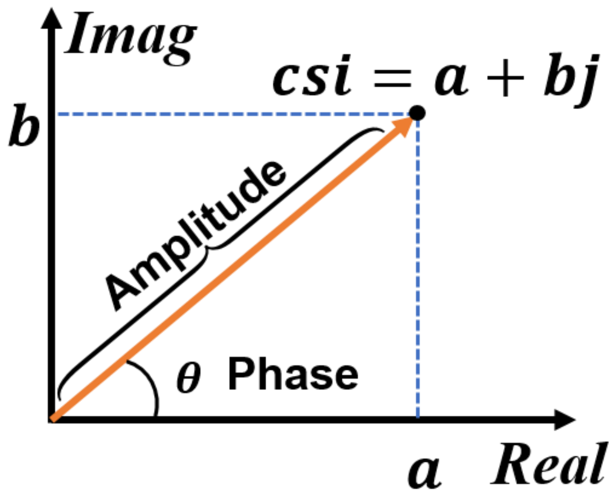

Reading taken at the CSE Department 2nd Floor (Room CSE-318)

Full data: csi_data

Thanks to Debjit Chatterjee for collecting these data and sharing the sample plots with me

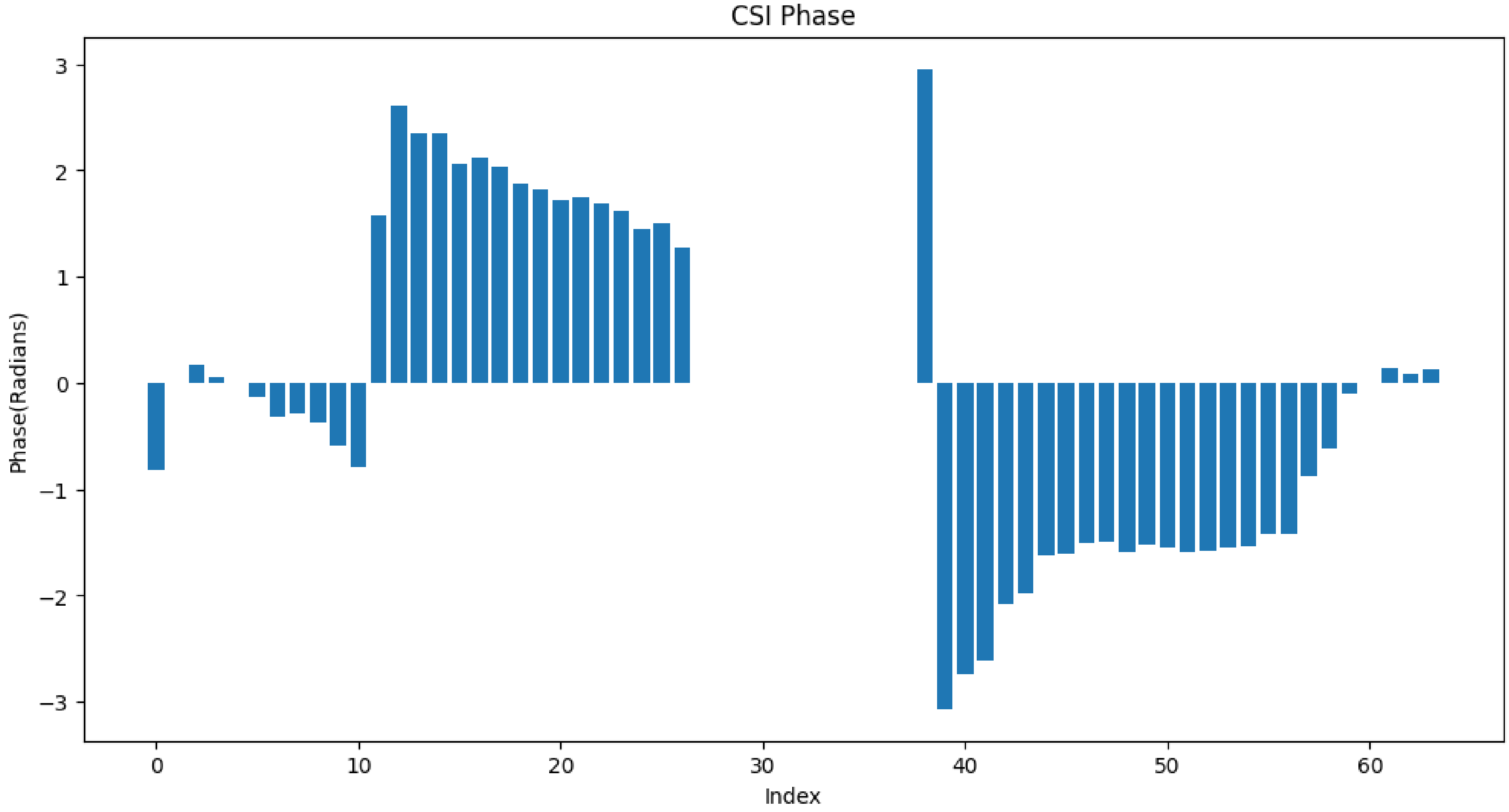
Computing Amplitude and Phase from the CSI Data

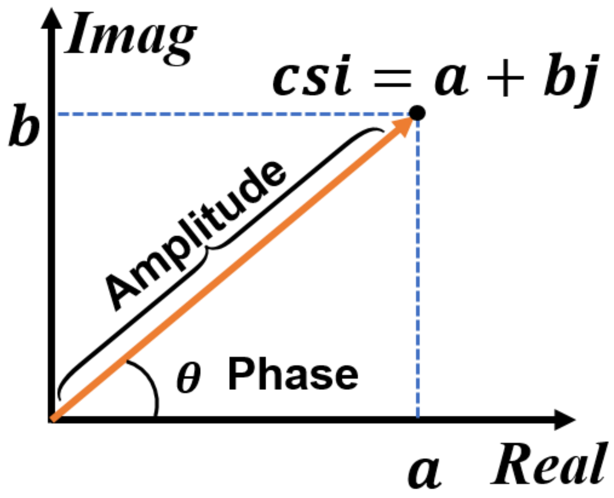




Thanks to Debjit Chatterjee for collecting these data and sharing the sample plots with me

Computing Amplitude and Phase from the CSI Data



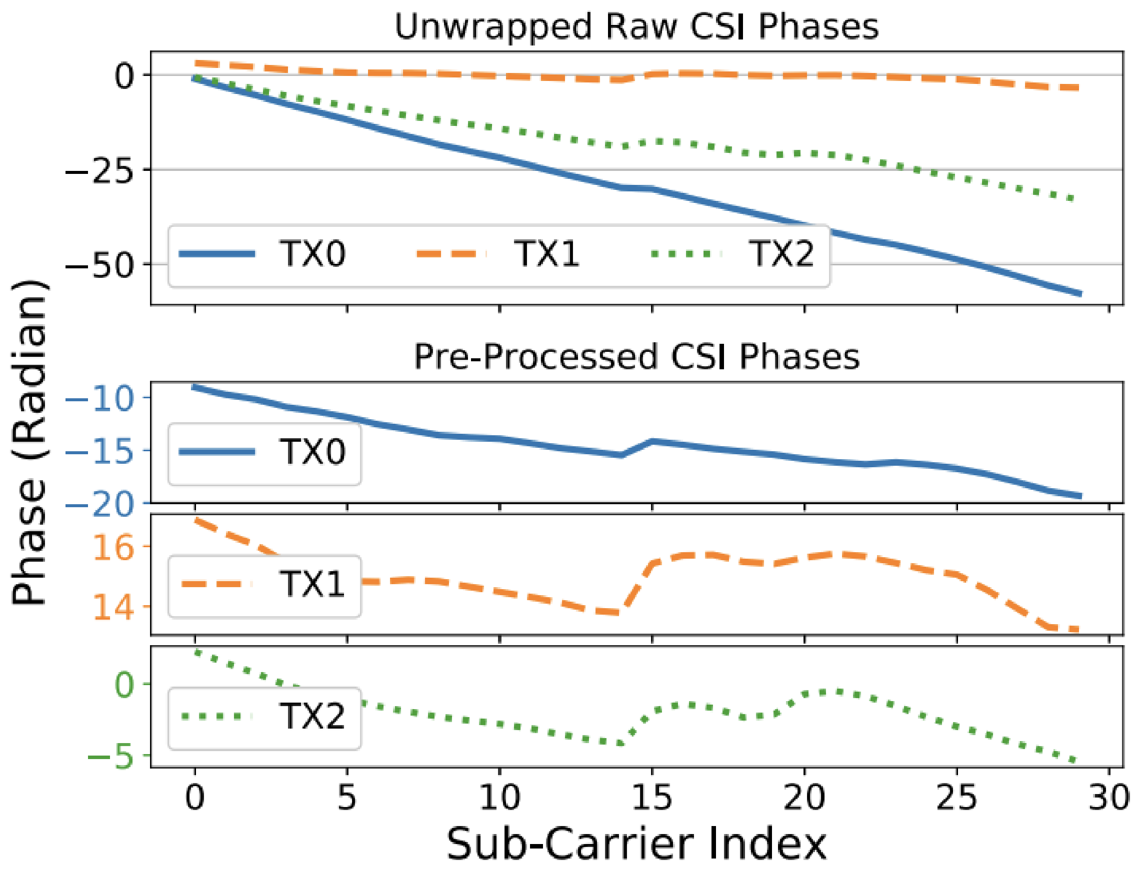


Thanks to Debjit Chatterjee for collecting these data and sharing the sample plots with me

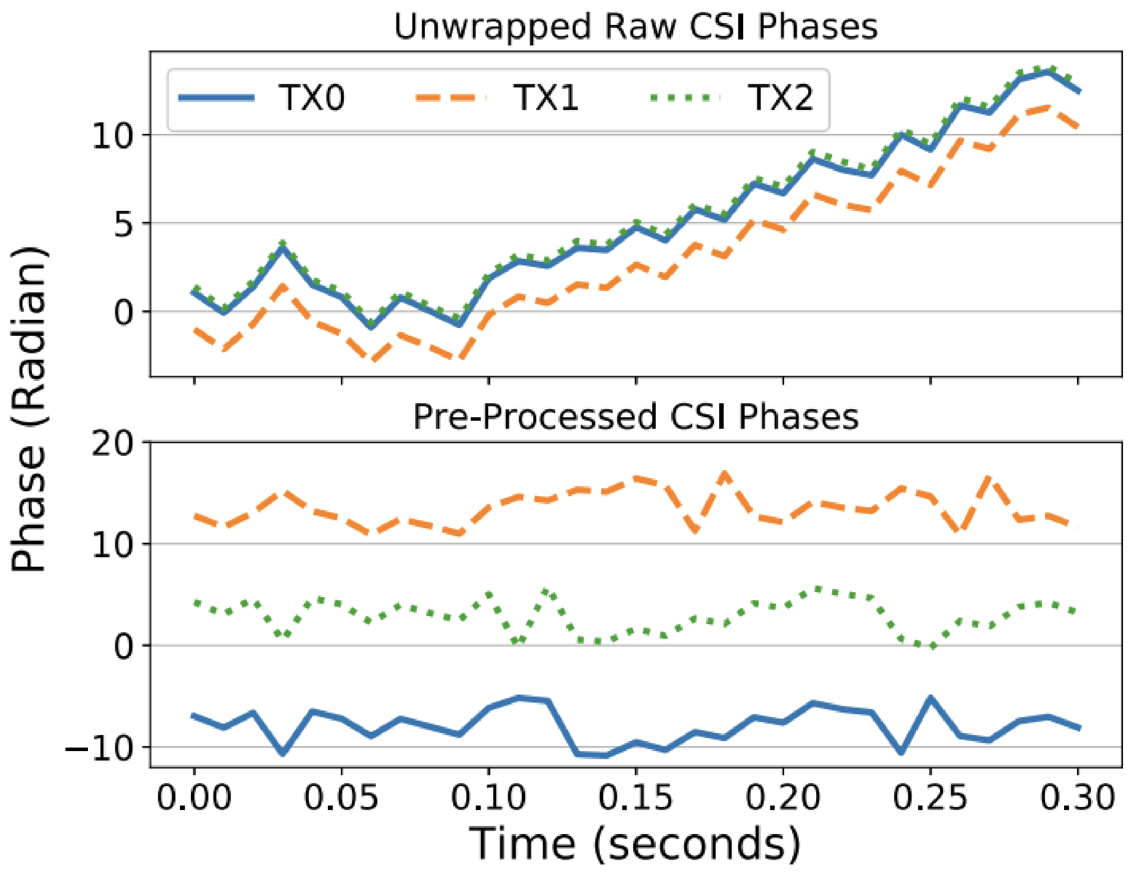
Signal Processing on the CSI Data

- **Phase Offset Removal**
  - Raw CSI measurements contain phase offsets due to hardware and software errors
  - Example: Sampling Time/Frequency Offset (STO/SFO) due to unsynchronized sampling of clocks/frequencies at the receiver and the transmitter
  - Assuming that the phase offset will be same across packets and subcarriers, use CSI phase difference of the adjacent subcarriers to average out the phase offset
    - *not accurate but useful for many sensing applications that are not impacted much by phase offsets (when you just need the pattern, say for activity recognition)*
  - Phase offset introduces estimation error for AoA and ToF computation – *need accurate values for localize and tracking applications*
    - Uses sophisticated approach like linear regression to estimate the phase offset

Impact of CSI Phase Offset



(a) CSI Phase vs. Subcarrier Index



(b) CSI Phase vs. Sampling Time

Source: https://dl.acm.org/doi/pdf/10.1145/3310194

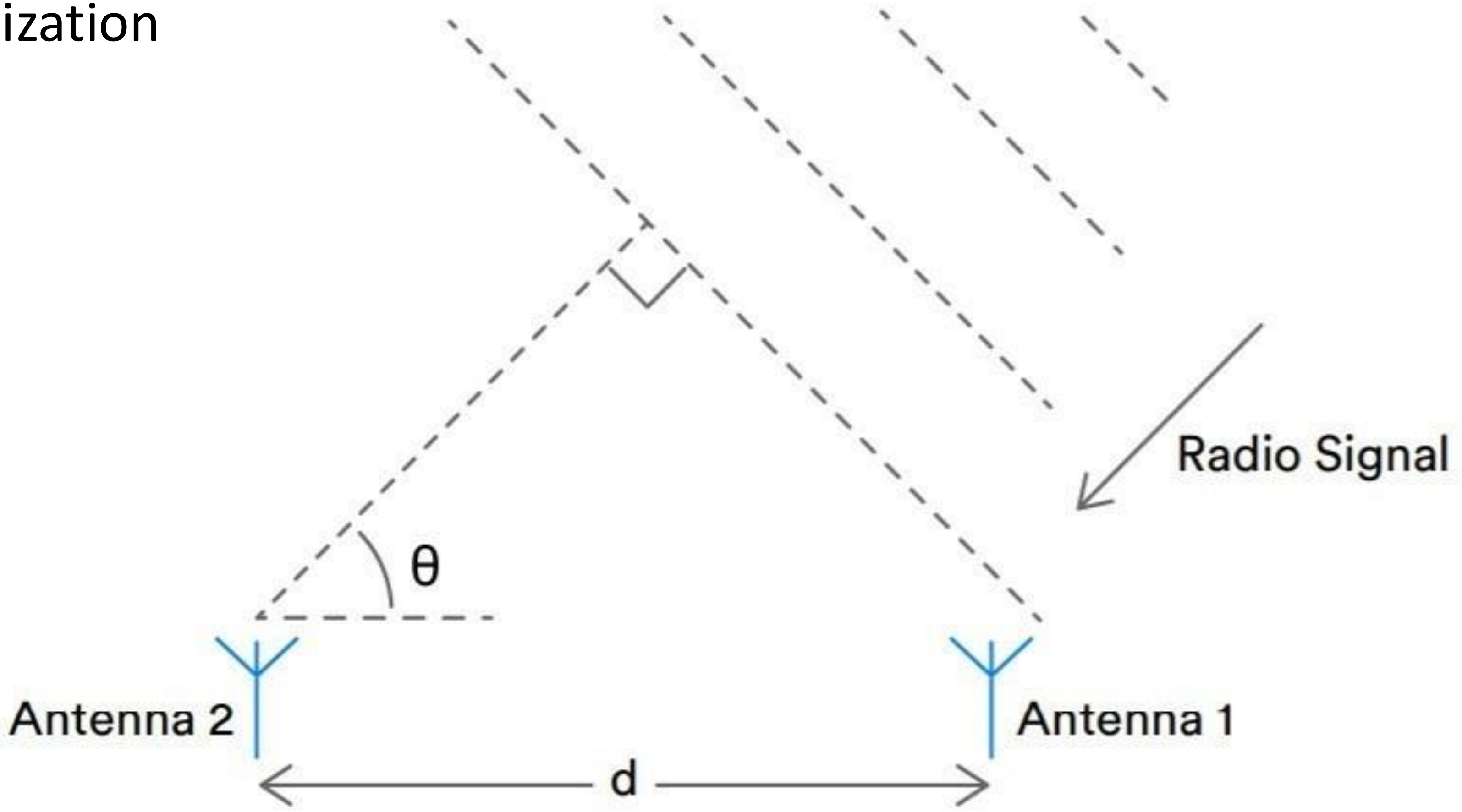
Signal Processing on the CSI Data

- **Outliers Removal**
  - Noises from the environment (particularly, high frequency noises) result outliers in the data – may affect the patterns
  - Low pass filters (LPF), Hampel filters, wavelet filters are typically used

- **Hardware/beamforming noise**
  - Methods like signal nulling are used to avoid beamforming noises
  - Cancels out the side-beams apart from the main beam

Angle of Arrival (AoA) Estimation

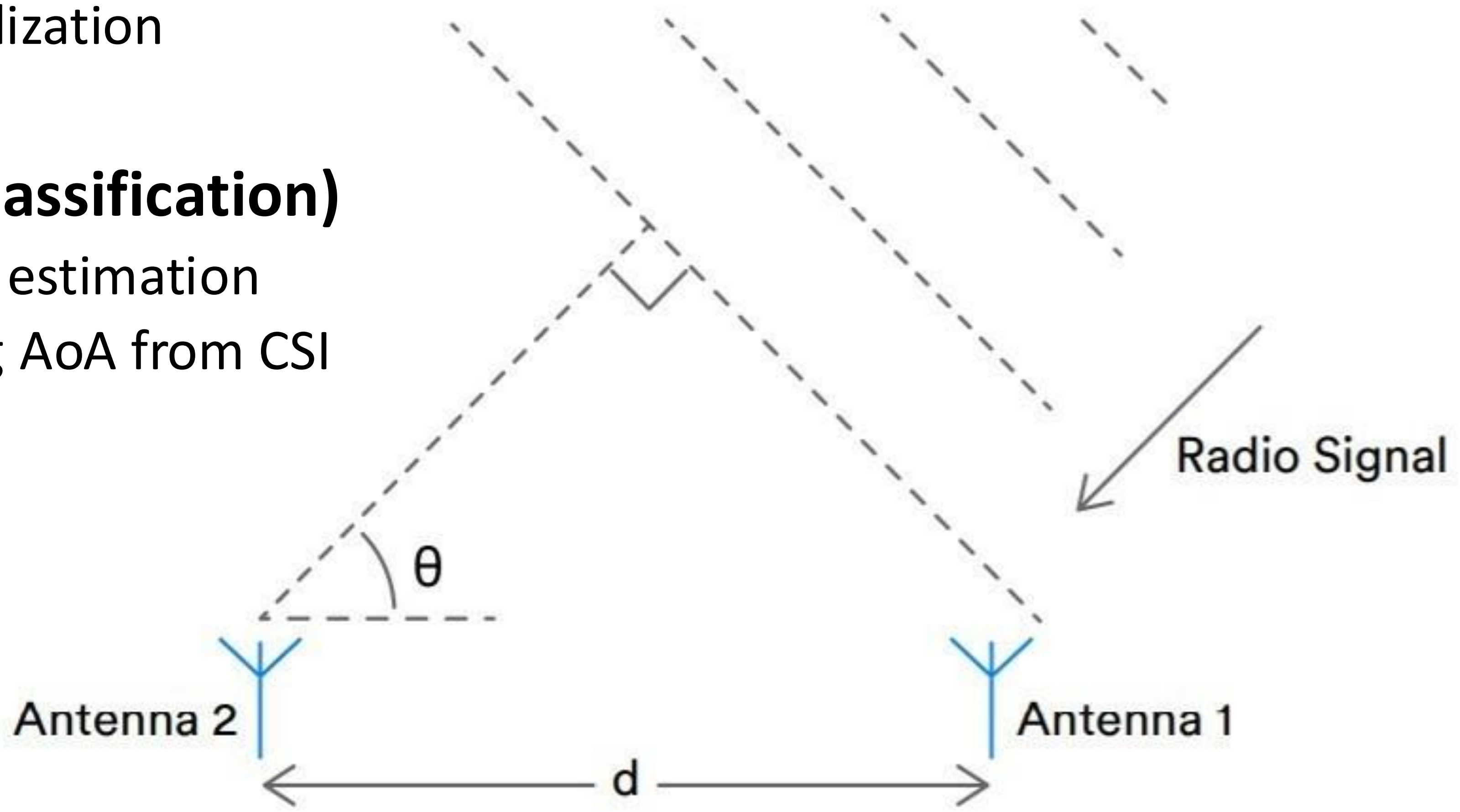
- Estimate the direction from which the signal arrives at a receiver from the transmitter
  - Used widely in indoor localization



Angle of Arrival (AoA) Estimation

- Estimate the direction from which the signal arrives at a receiver from the transmitter
  - Used widely in indoor localization

- **MUSIC (Multiple Signal Classification)**
  - A popular method for AoA estimation
  - Used widely for computing AoA from CSI



Multiple Signal Classification (MUSIC)

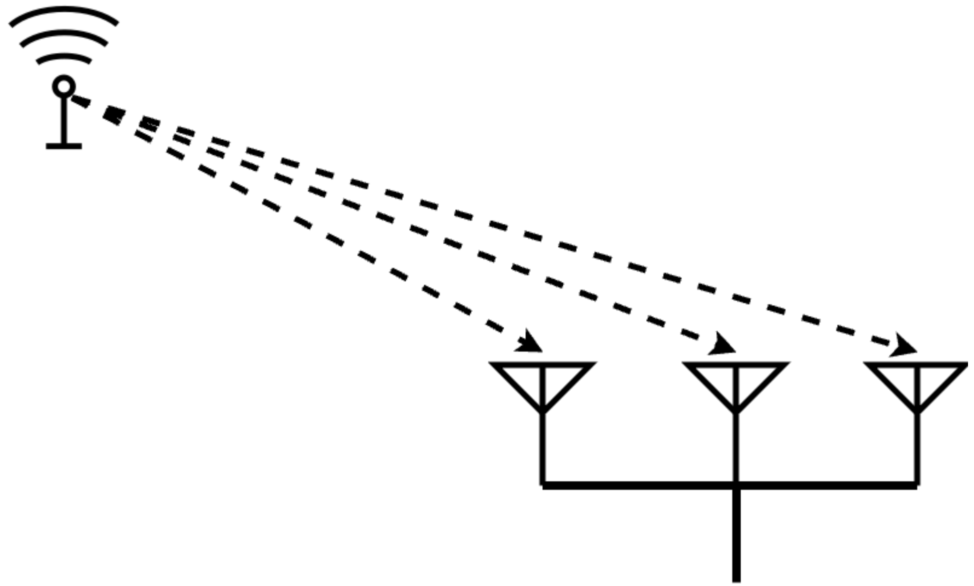
- Leverages the structure of the received signal at an antenna array
  - Separates the signal subspace (where the actual signal lies) from the noise subspace
  - Estimates AoA by finding directions that corresponds to the smallest projection onto the noise subspace
- **Signal subspace:** Components of the received data that are directly influenced by the incoming signal
- **Noise subspace:** Portion of the data that corresponds to noise and interference (noise is typically weaker and uncorrelated across the antennas)
- **The noise subspace is orthogonal to the signal subspace –** key property used in the MUSIC algorithm
  - The signal should be sufficiently different from the noise to help the receiver to separate the signal from the noise

Disclaimer: *ChatGPT 4o* has been used to generate some explanations of the MUSIC algorithm

Multiple Signal Classification (MUSIC)

- Let H be the input CSI matrix; the MUSIC algorithm works as follows.
- **Covariance Matrix Calculation**
  - MUSIC first computes the covariance matrix $R=HH^H$, where H^H is the Hermitian transpose (complex conjugate transpose) of the CSI matrix H
  - Captures the spatial correlation between the signal received at different antennas

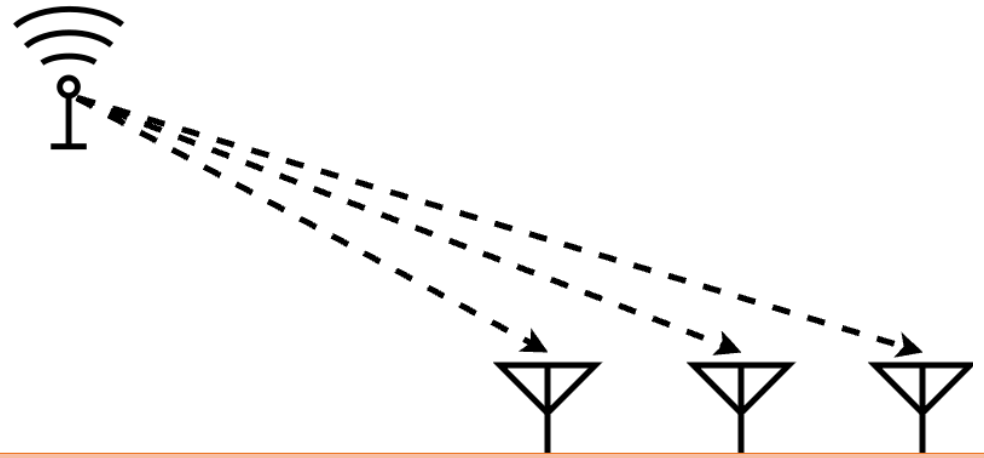
Note that whenever a signal comes from a specific direction, it reaches the antennas with phase shifts and attenuation based on the geometry of the antenna array



Multiple Signal Classification (MUSIC)

- Let H be the input CSI matrix; the MUSIC algorithm works as follows.
- **Covariance Matrix Calculation**
  - MUSIC first computes the covariance matrix $R=HH^H$, where H^H is the Hermitian transpose (complex conjugate transpose) of the CSI matrix H
  - Captures the spatial correlation between the signal received at different antennas

Note that whenever a signal comes from a specific direction, it reaches the antennas with phase shifts and attenuation based on the geometry of the antenna array



Leads to predictable variance of signals across antennas

Multiple Signal Classification (MUSIC)

- Let H be the input CSI matrix; the MUSIC algorithm works as follows.
- **Covariance Matrix Calculation**
  - MUSIC first computes the covariance matrix $R=HH^H$, where H^H is the Hermitian transpose (complex conjugate transpose) of the CSI matrix H
  - Captures the spatial correlation between the signal received at different antennas
- In practice, the covariance matrix is computed as $R=E[HH^H]$, where $E[.]$ is the expectation operator that computes the average over multiple measurements

Multiple Signal Classification (MUSIC)

- Let H be the input CSI matrix; the MUSIC algorithm works as follows.
- **Covariance Matrix Calculation**
  - MUSIC first computes the covariance matrix $R=HH^H$, where H^H is the Hermitian transpose (complex conjugate transpose) of the CSI matrix H
  - Captures the spatial correlation between the signal received at different antennas
- In practice, the covariance matrix is computed as $R=E[HH^H]$, where $E[.]$ is the expectation operator that computes the average over multiple measurements
- Notably, Signals coming from the same source (or direction) will induce similar patterns of phase shifts and amplitudes across the antennas
  - These similarities show up as correlations between the signals received at different antennas.

Multiple Signal Classification (MUSIC)

- **Eigenvalue Decomposition**
  - The covariance matrix R is decomposed into eigenvalue and eigenvectors:

$$R = E \Lambda E^H$$

    E is the matrix of eigenvectors
    Λ is the diagonal matrix of eigenvalues, sorted in descending order

$$\boldsymbol{Av = \lambda v}$$

A is the matrix, v is the eigenvector, and λ is the eigenvalue
Note: Eigenvectors are often viewed as the "*principal axes*" of transformation. When a matrix is applied to a space (like scaling, stretching, or rotating vectors), the eigenvectors are the directions along which these transformations occur without altering direction.

What Eigenvectors Represent in MUSIC

- The eigenvectors of the covariance matrix represent the principal directions in the signal space where the energy or correlation of the received signals is strongest.
- **Signal Subspace**: The eigenvectors corresponding to the largest eigenvalues span the signal subspace.
  - These eigenvectors represent the directions of arrival (AoA) of the incoming signals because they capture the dominant signal components in the received data.
- **Noise Subspace**: The eigenvectors corresponding to the smallest eigenvalues span the noise subspace.
  - These eigenvectors capture the directions that are primarily noise, where no significant signal energy is present.

What Eigenvalues Represent in MUSIC

- The eigenvalues represent the **strength or power** of the signal components along the corresponding eigenvector directions.
- Large eigenvalues correspond to directions where the signal energy is concentrated
  - The directions from which the actual signal is arriving (the signal subspace).
- Small eigenvalues correspond to noise
  - Note that the noise power is typically much smaller than the signal power (the noise subspace).

Orthonormal Basis of Eigenvectors

- The eigenvectors form an **orthonormal basis**, which means they are **mutually orthogonal**
  - a property of eigenvalue decomposition when dealing with Hermitian (or symmetric) matrices like the covariance matrix R
- Therefore,
  - Eigenvectors associated with different eigenvalues are orthogonal to each other
  - The signal and noise subspaces correspond to different sets of eigenvectors, so they are orthogonal by construction

Interpretation of Orthogonality

- Orthogonality between the signal and noise subspaces
  - A vector in the signal subspace (e.g., the steering vector for the actual signal's AoA) will have no component in the noise subspace

- Conversely, the steering vectors for angles that do not correspond to an actual signal AoA will have non-zero projections onto the noise subspace

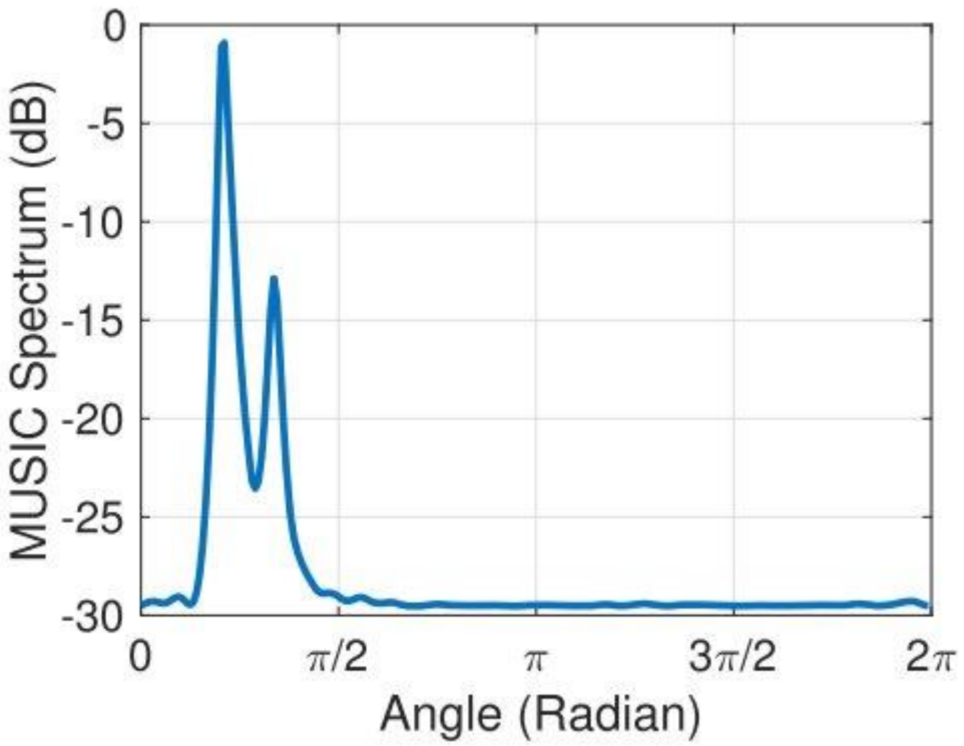
Exploiting Orthogonality in MUSIC

- The MUSIC algorithm uses this orthogonality property to estimate the AoA.
  - For a hypothesized angle θ, a steering vector $a(\theta)$ is projected onto the noise subspace.

- If the projection of $a(\theta)$ onto the noise subspace is **small**
  - the steering vector is nearly orthogonal to the noise subspace
  - that angle is likely the true AoA, since the steering vector lies in the signal subspace

- If the projection is large, the hypothesized angle does not correspond to an actual signal direction.

AoA Estimation by MUSIC



The Core Idea: The signal components dominate certain directions (eigenvectors), and noise is uncorrelated and spreads across others.

Estimating Time of Flight (ToF)

- ToF is the time it takes for a signal to travel from the transmitter to the receiver

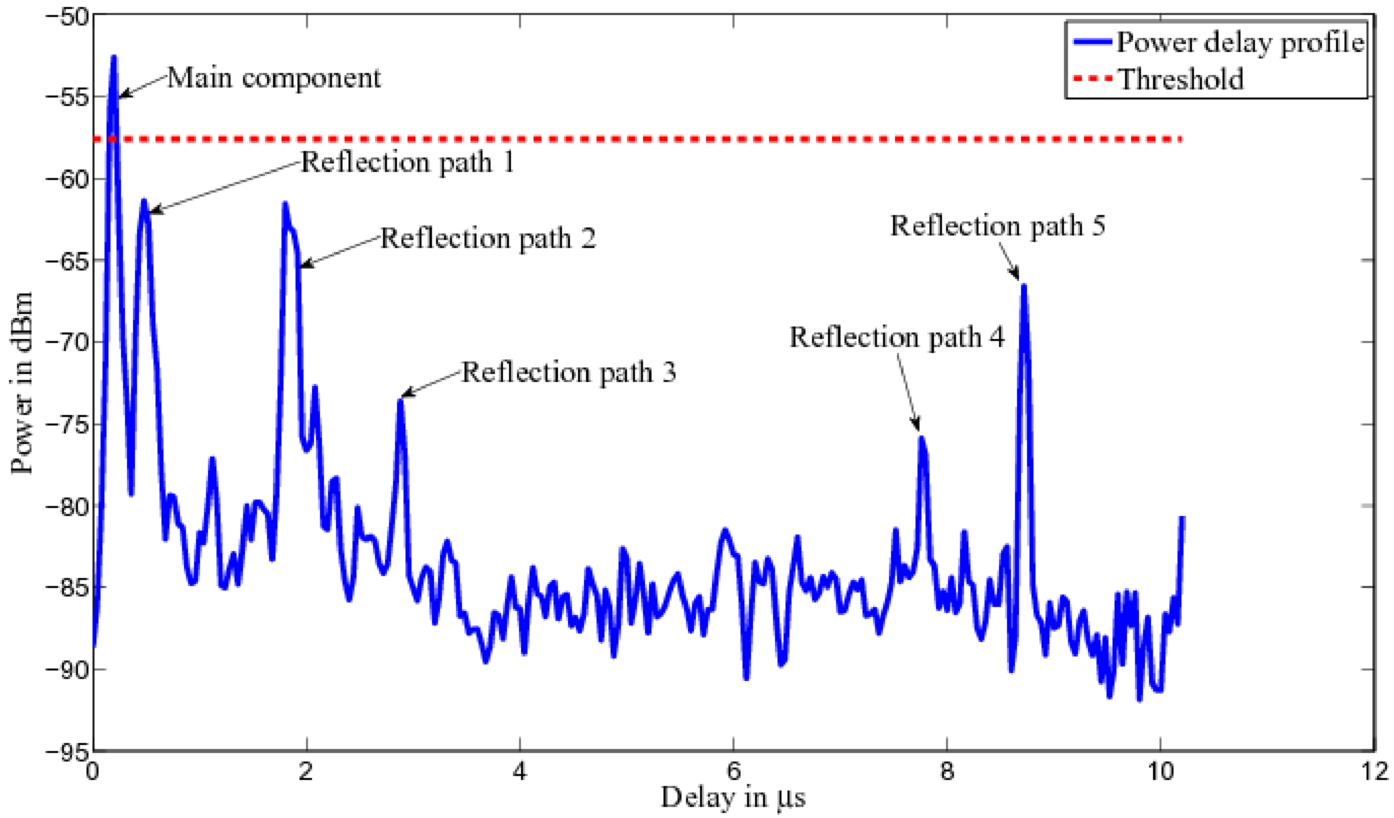
- The **Power-Delay Profile** (PDP) is computed from the CSI, which is used in ToF estimation
    - PDP helps to estimate the delay of signals as they travel through different paths between the transmitter and receiver

Power-Delay Profile (PDP)

- A representation of the power of a signal as a function of time delay
  - Shows how much signal power arrives at the receiver at different delays, which correspond to different propagation paths (e.g., the direct path and various reflected paths)
  - Maps the distribution of signal energy over time

Power-Delay Profile (PDP)

- For a multipath environment, a transmitted signal takes multiple paths to reach the receiver
  - Each path will introduce a different delay due to varying distances
  - PDP captures these delays by showing which parts of the signal energy arrive at the receiver at what times



Computing PDP from CSI

- Note that, CSI data provides the channel frequency response *H(f)* for each subcarrier *f*

- We first apply an *Inverse Discrete Fourier Transform* (IDFT) to the CSI data

$$h(t) = IDFT(H(f))$$

  - Transforms the CSI data from the frequency domain to the time domain (or delay domain
  - Result is a time-domain representation of the channel, called as the **channel impulse response (CIR)**.

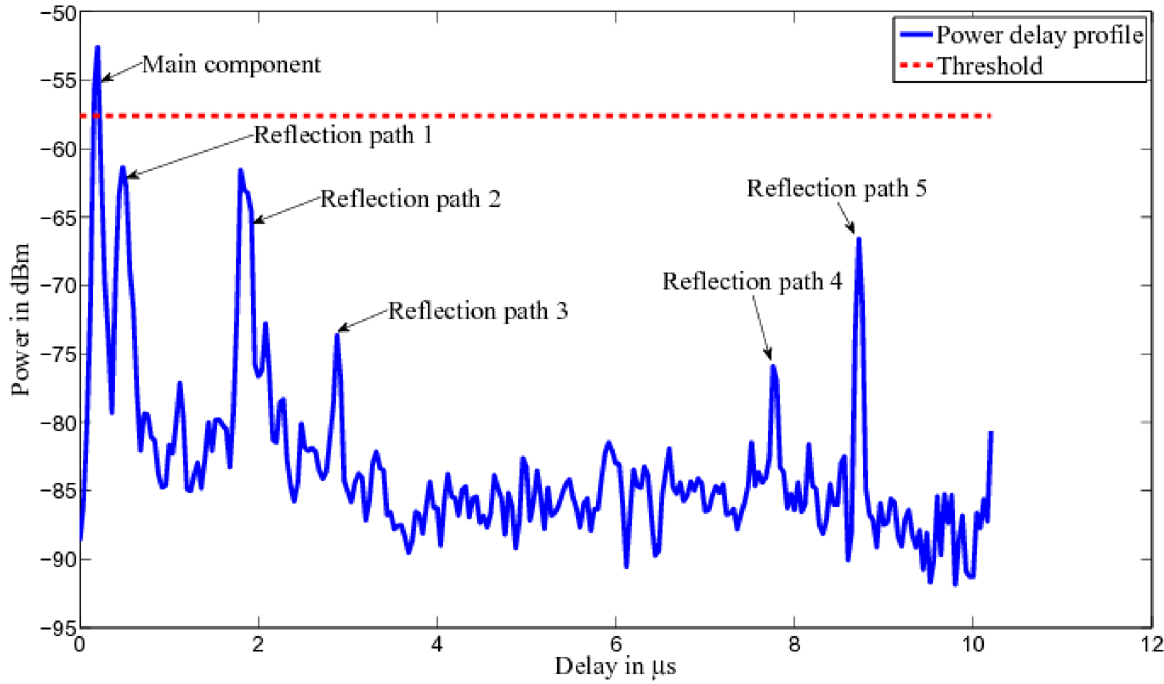
Power Calculation

- PDP is obtained by squaring the magnitude of the channel impulse response

$$PDP(t) = |h(t)|^2$$

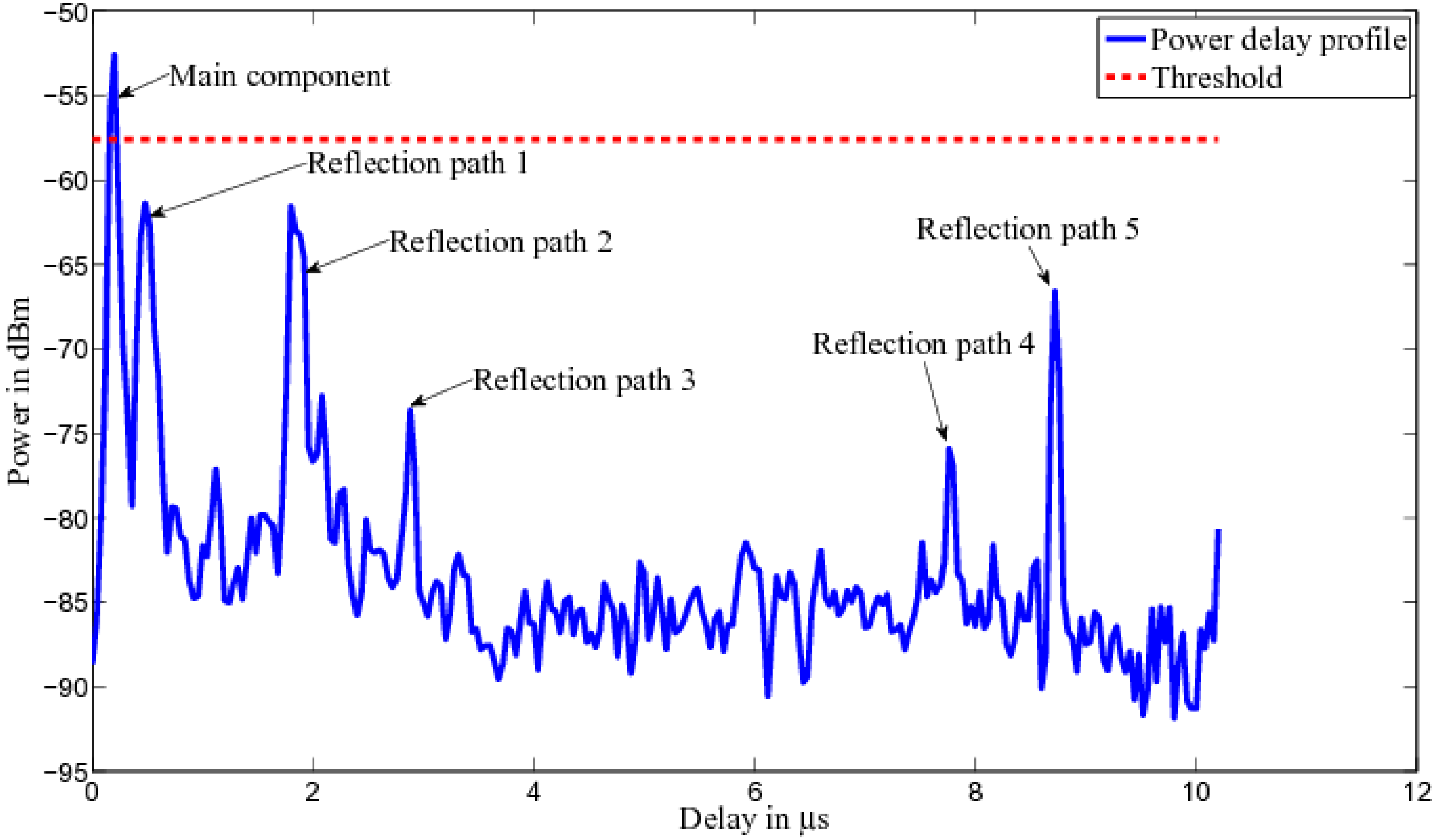
  - Gives the power of the received signal as a function of time delay.

- Peaks in the PDP indicate the presence of significant signal paths, corresponding to different propagation delays.



Using PDP for ToF Estimation

- ToF can be estimated from the PDP by identifying the time delay of the first significant peak, which usually corresponds to the direct path between the transmitter and receiver



Using PDP for ToF Estimation

- ToF can be estimated from the PDP by identifying the time delay of the first significant peak, which usually corresponds to the direct path between the transmitter and receiver

- In a multipath environment, the signal can travel directly from the transmitter to the receiver (the line-of-sight path) or via reflections off walls, objects, etc.
    - The ToF corresponding to the direct path (**the first peak in the PDP**) is the shortest delay.

- Other peaks in the PDP represent signal paths that include reflections or scatterings, which take longer to reach the receiver.

Using PDP for ToF Estimation

- Once the ToF is estimated from the first peak in the PDP, the distance between the transmitter and receiver can be computed using the speed of light *c*:

$$d = ToF \times c$$

- Note, low SNR can make it harder to detect the first peak in the PDP
  - Need sophisticated algorithm for peak detection

Using PDP for ToF Estimation

- Once the ToF is estimated from the first peak in the PDP, the distance between the transmitter and receiver can be computed using the speed of light *c*:

$$d = ToF \times c$$

- Note, low SNR can make it harder to detect the first peak in the PDP
  - Need sophisticated algorithm for peak detection

We shall next see how CSI can be used for developing interesting WiFi sensing applications

WiTrack: Motion Tracking via Radio Reflections off the Body

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.

1 Introduction

Recent years have witnessed a surge in motion tracking and localization systems. Multiple advances have been made both in terms of accuracy and robustness. In particular, RF localization using WiFi and other communication devices has reached sub-meter accuracy and demonstrated its ability to deal with occlusions and non-line of sight scenarios [31, 18]. Yet these systems require the user to carry a wireless device in order to be localized. In contrast, systems like Kinect and depth imaging have revolutionized the field of human-computer interaction by enabling 3D motion tracking without instrumenting the body of the user. However, Kinect and imaging systems require a user to stay within the device's line-of-sight and

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 technique called FMCW (frequency modulated carrier wave) which maps differences in time to shifts in the carrier frequency; such frequency shifts are easy to measure in radio systems by looking at the spectrum of the received signal.

A second challenge stems from multipath effects, which create errors in mapping the delay of a reflection to the distance from the target. WiTrack has to deal with two types of multipath effects. Some multipath effects are due to the transmitted signal being reflected off walls and furniture. Others are caused by the signal first reflecting off the human body then reflecting off other objects. This is further complicated by the fact that in non-line-of-sight settings, the strongest signal is not the one directly bouncing off the human body. Rather it is the signal that avoids the occluding object by bouncing off some side walls

NSDI 2024

Slides are based on the presentation by the original authors, available in the above link

Can We See through Walls with Wireless Signals?

- Tracks the 3D motion of a user using radio reflections that bounce off her body
- Works through walls and occlusions, but does not require the user to carry any wireless device
- The user lift her hand and point at objects in the environment; the device detects the direction of the hand motion, enabling the user to identify objects of interest

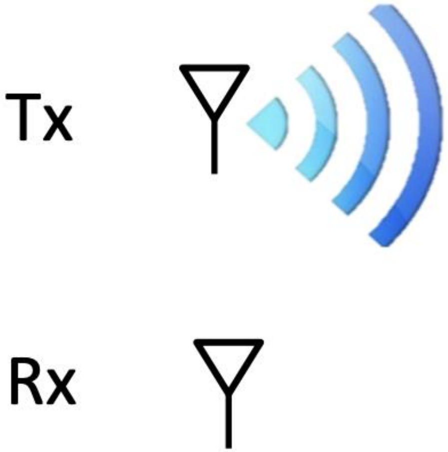
WiTrack: The Core Idea

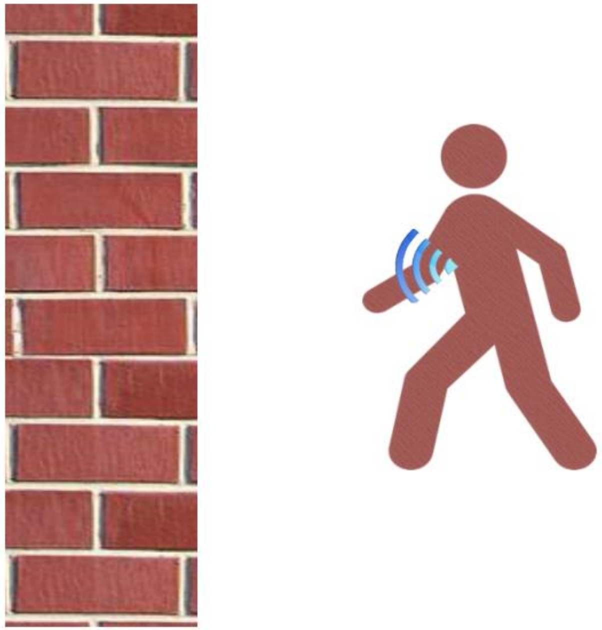
- Use a transmitter and a receiver to estimate the signal reflection time from the body
  - Actually uses THREE Rx antennas – *why, we'll see later!*

- Measures the time it takes for its signal to travel from the transmit antenna to the reflecting body, and then back to each of the receive antennas – *this time is called the Time of Flight (TOF)*.

- Once it obtains the TOF as perceived from each of its receiving antennas, WiTrack leverages the geometric placement of its antennas to localize the moving body in 3D

Measuring Distances

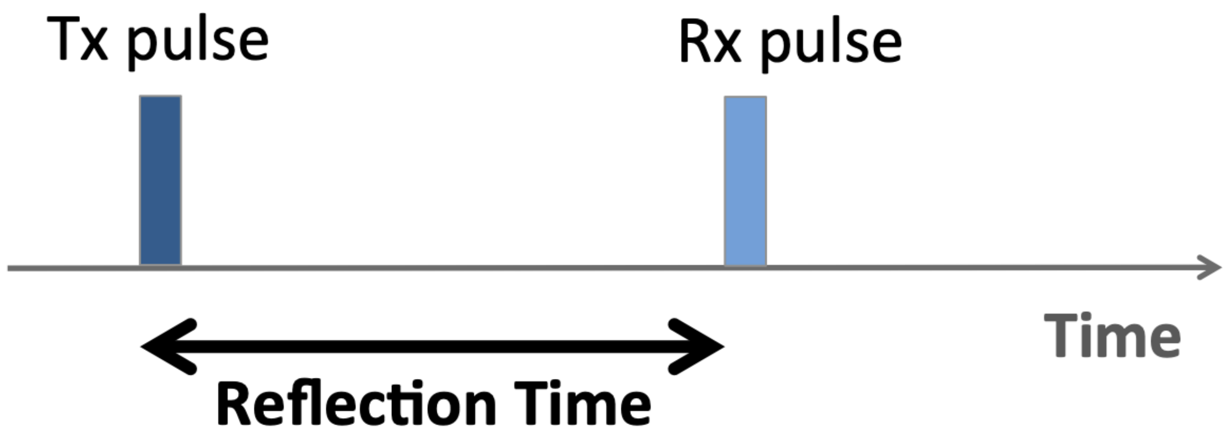




Distance = Reflection Time x Speed of Light

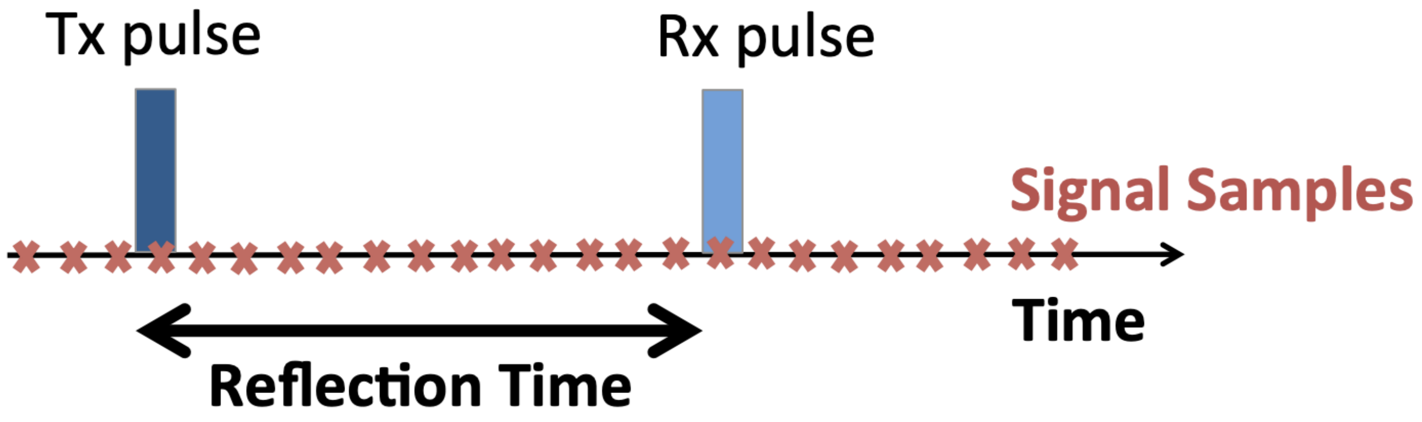
How Do We Measure the Reflection Time?

- **Option 1:** Transmit short pulses and then listen for an echo



How Do We Measure the Reflection Time?

- **Option 1:** Transmit short pulses and then listen for an echo

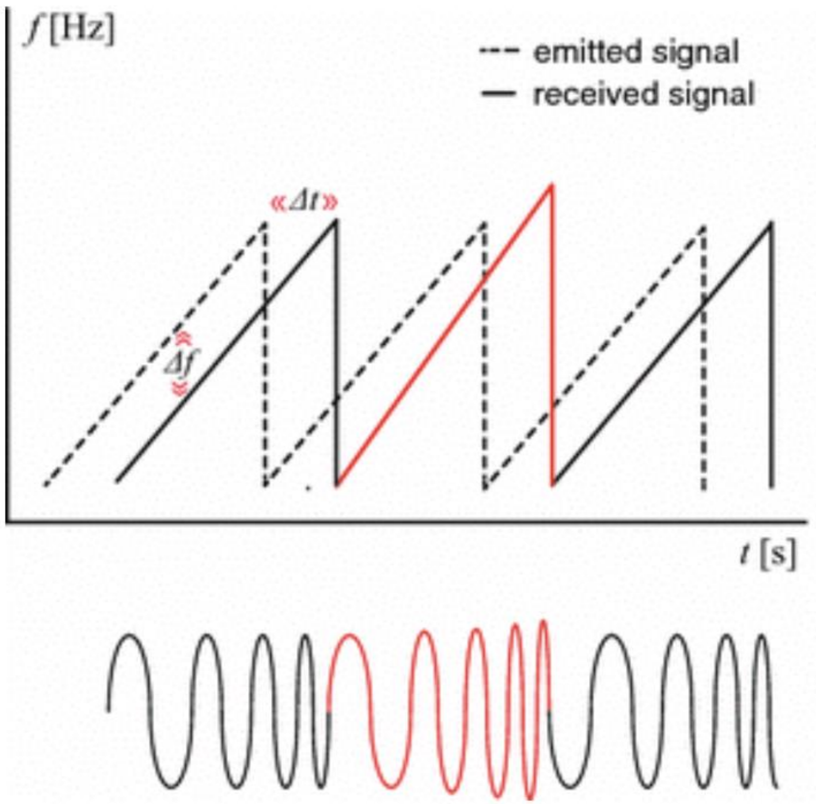


Capturing the pulses need sub-nanosecond sampling !

Multi-GHz samplers are expensive and have high noise

How Do We Measure the Reflection Time?

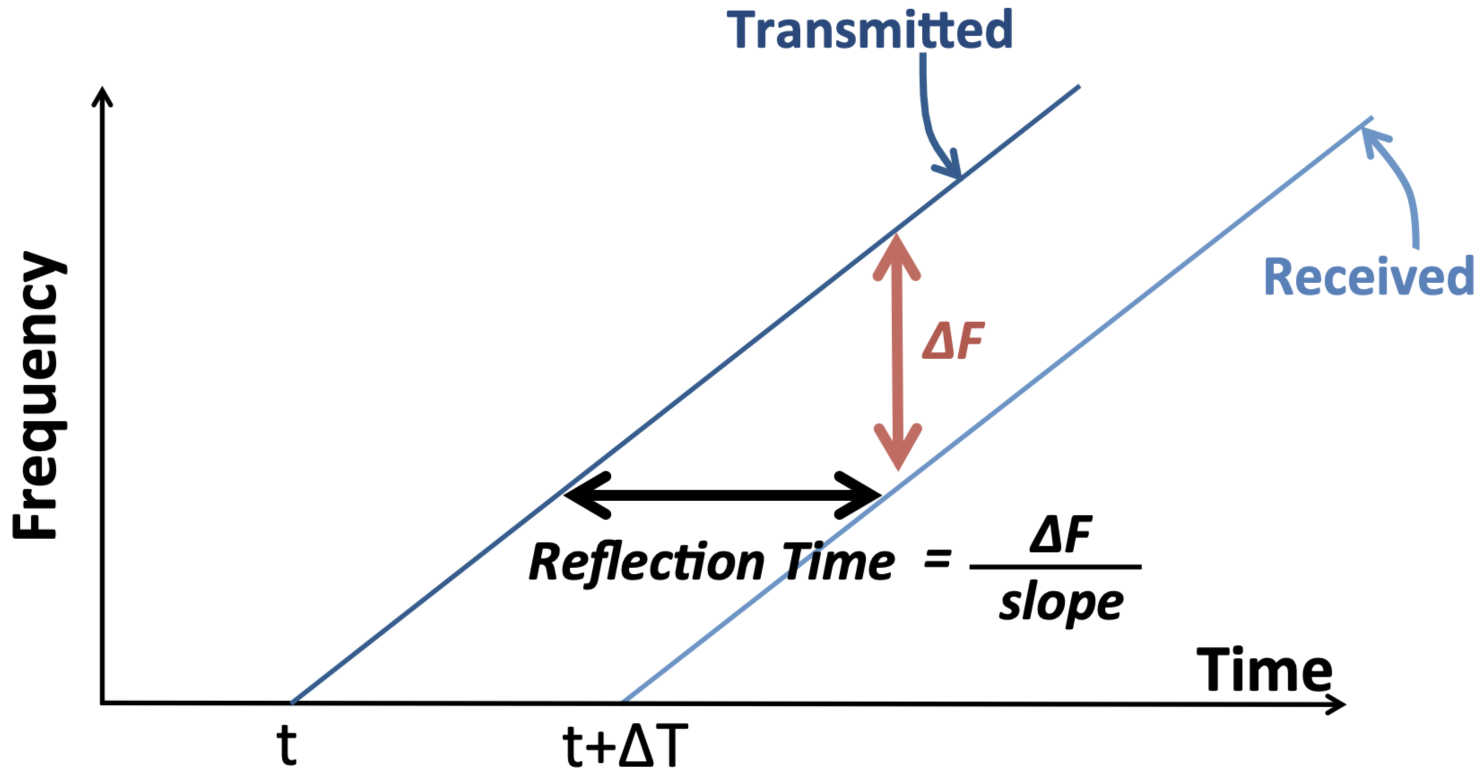
- **Option 2**: Use frequency modulated continuous wave (FMCW) signals
  - Transmits a narrowband signal (e.g., a few KHz) whose carrier frequency changes linearly with time
  - Used widely in radars (*we'll see many usage later on)*



- To identify the distance from a reflector, FMWC compares the carrier frequency of the reflected signal to that of the transmitted signal
  - Since the carrier frequency is changing linearly in time, delays in the reflected signals translate into frequency shifts in comparison to the transmitted wave

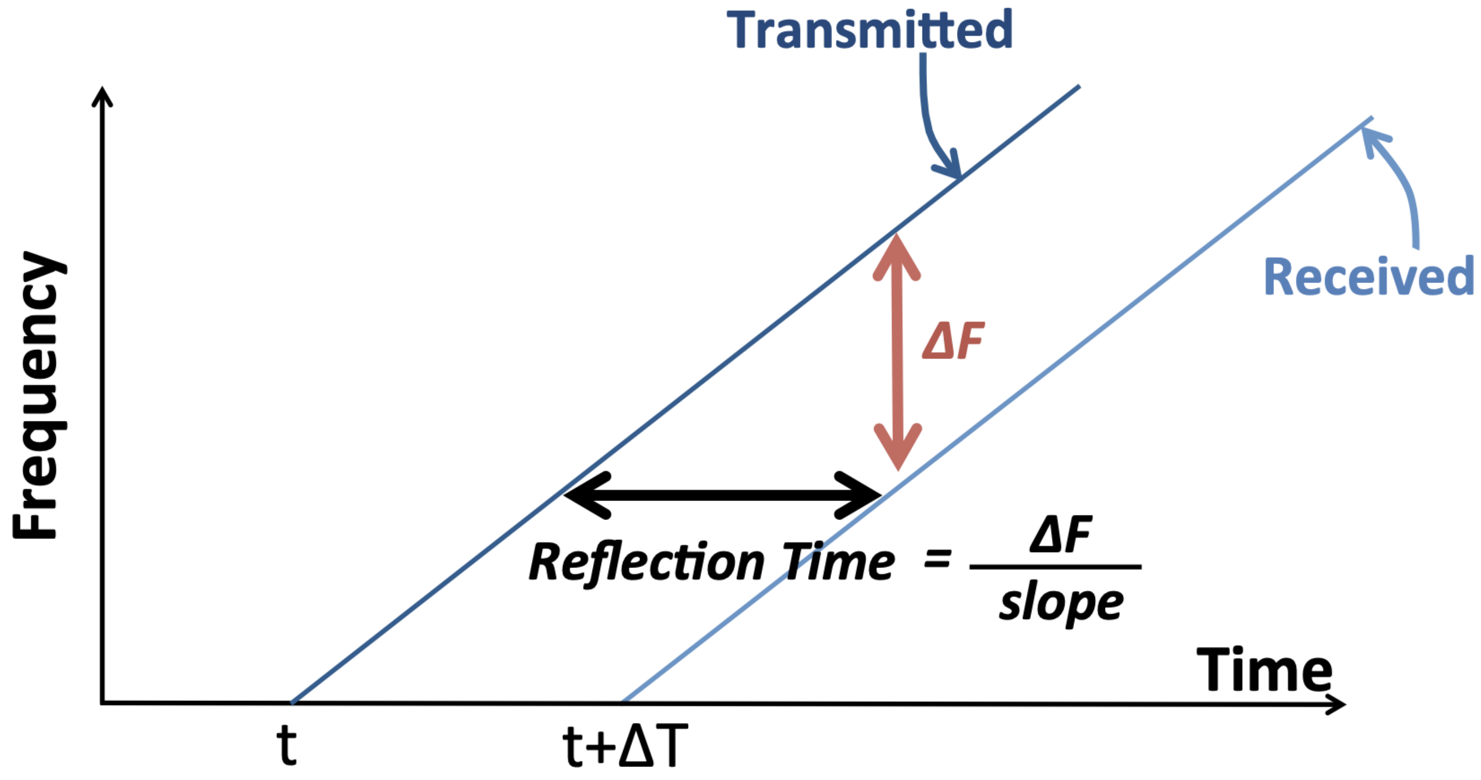
How Do We Measure the Reflection Time?

- **Option 2**: Use frequency modulated continuous wave (FMCW) signals



How Do We Measure the Reflection Time?

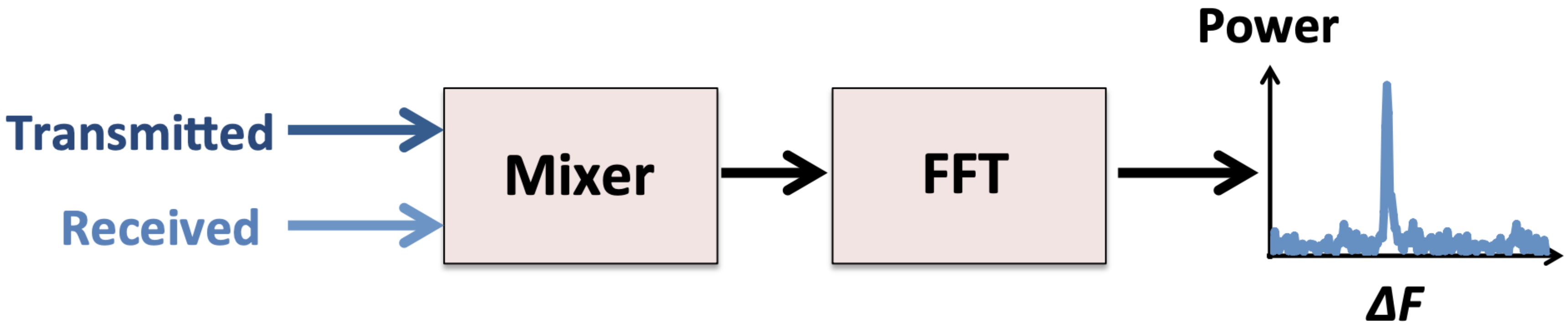
- **Option 2**: Use frequency modulated continuous wave (FMCW) signals



How do we measure ΔF?

Measuring ΔF

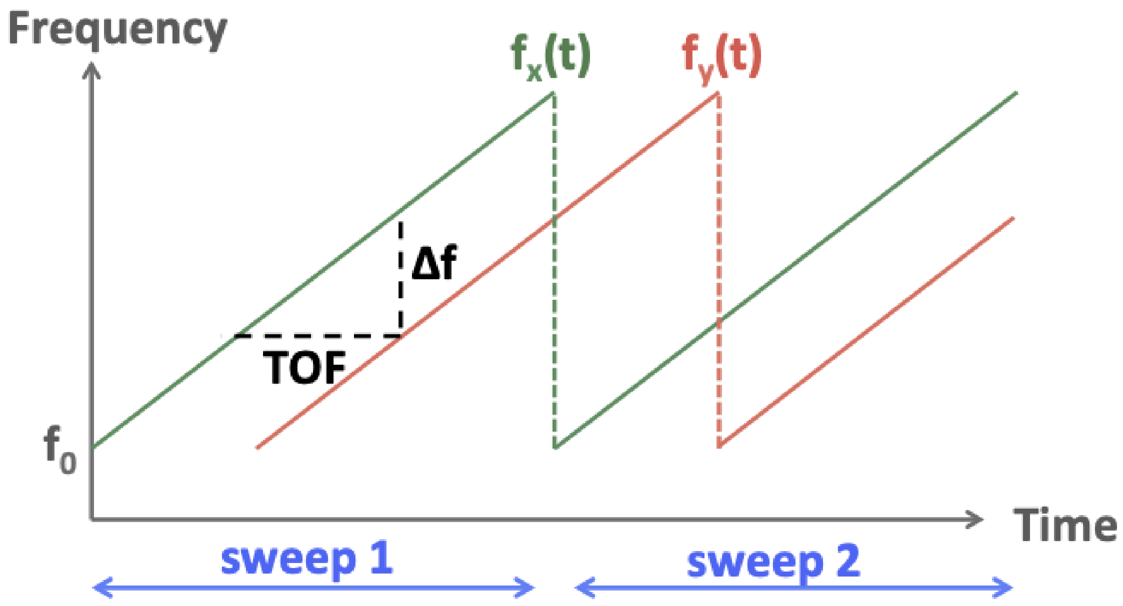
- Subtracting frequencies is easy
  - For example, removing carrier in WiFi
  - Can be done using a mixer (low-power and cheap)



- We can compute the TOF from ΔF, and then use it to compute the distance

Implementing FMCW for WiTrack

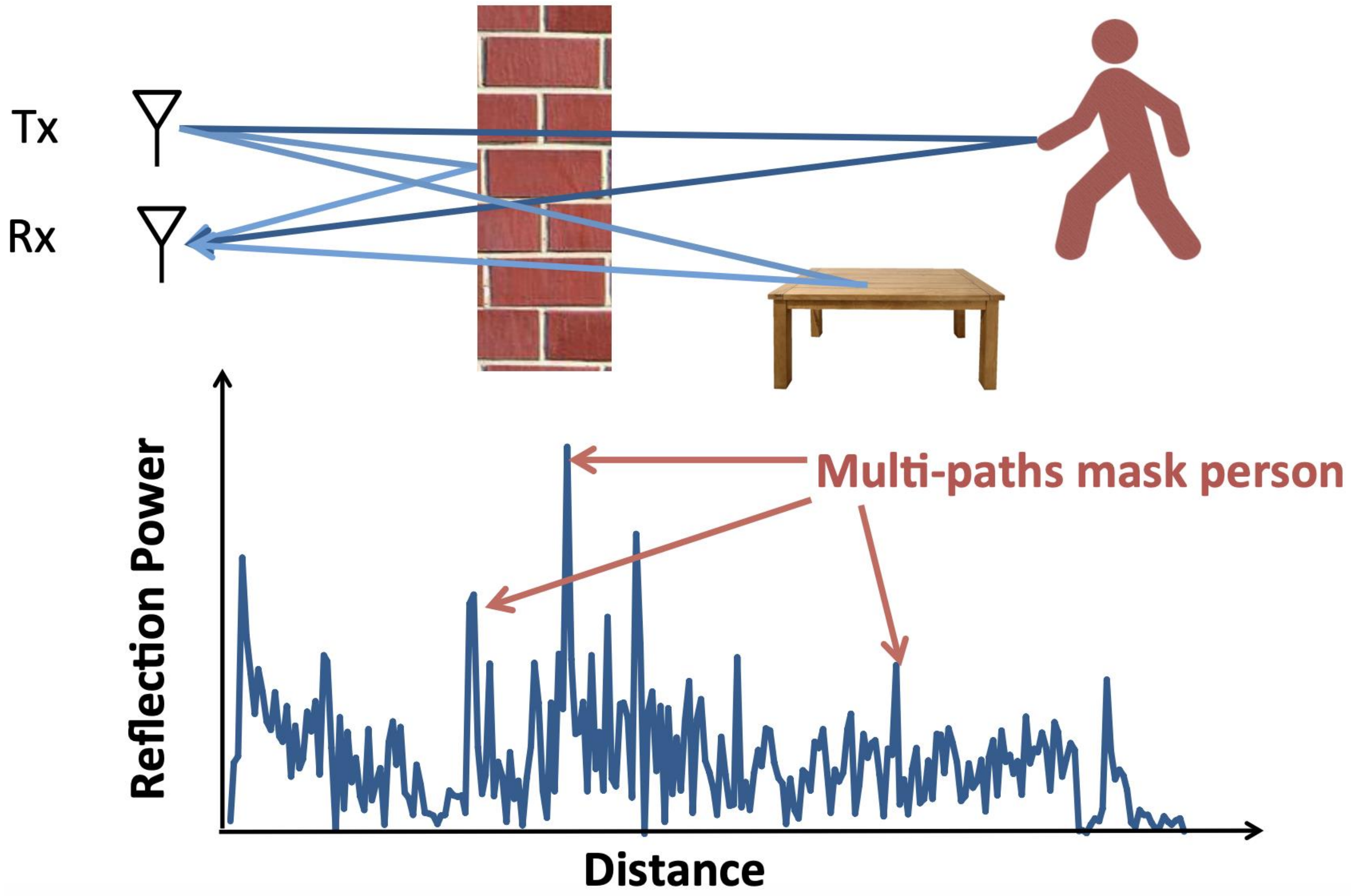
- In practice, use multiple sweeps over time
- The resolution of an FMCW system is a function of the total bandwidth that the carrier frequency sweeps
    - Ability to distinguish between two nearby locations, which depends on the ability to distinguish their TOFs, which itself depends on the resolution in distinguishing frequency shifts Δf



- The FFT is typically taken over a duration of one sweep of the carrier frequency

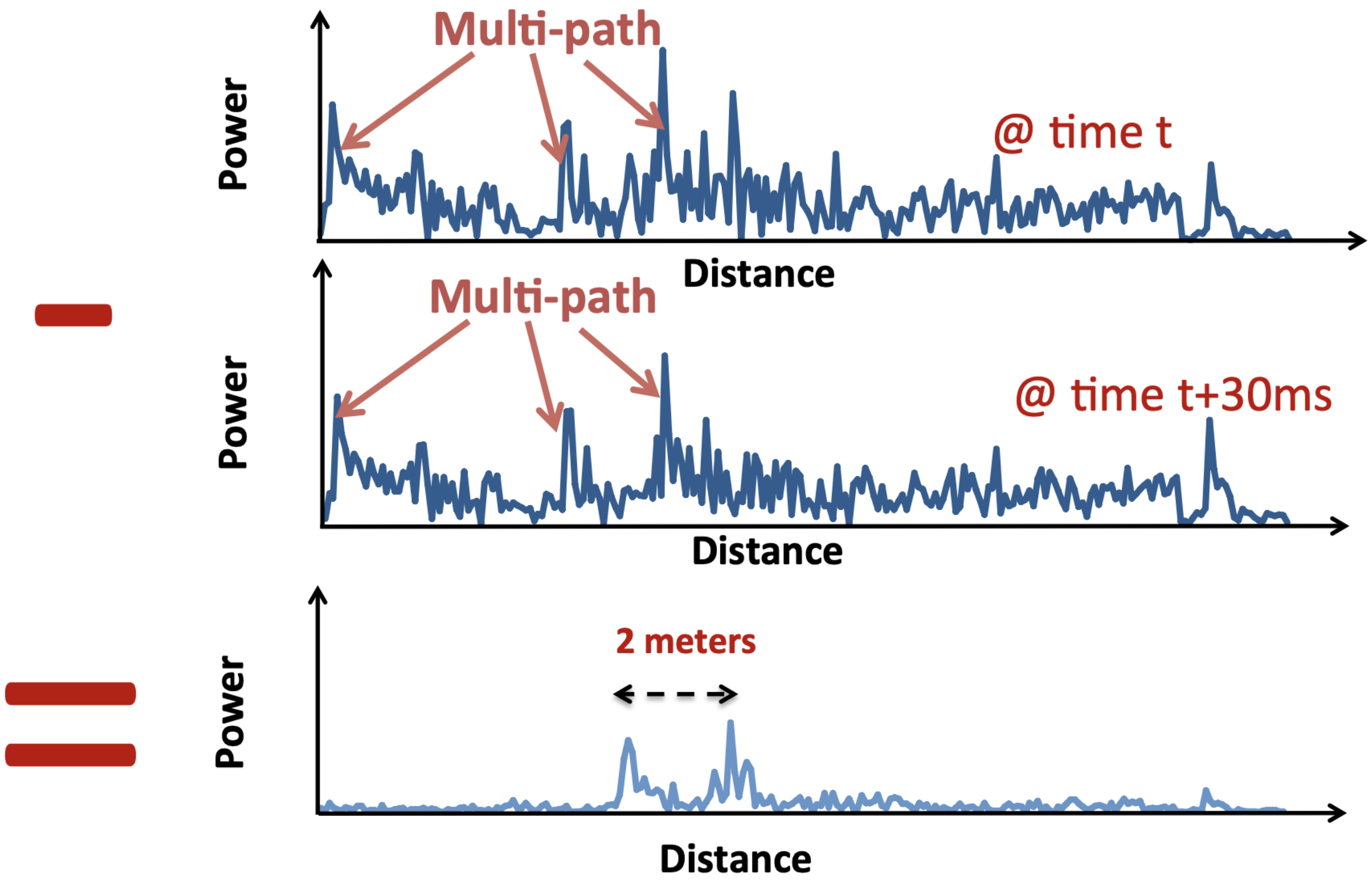
Challange

- Multipath introduces many reflections



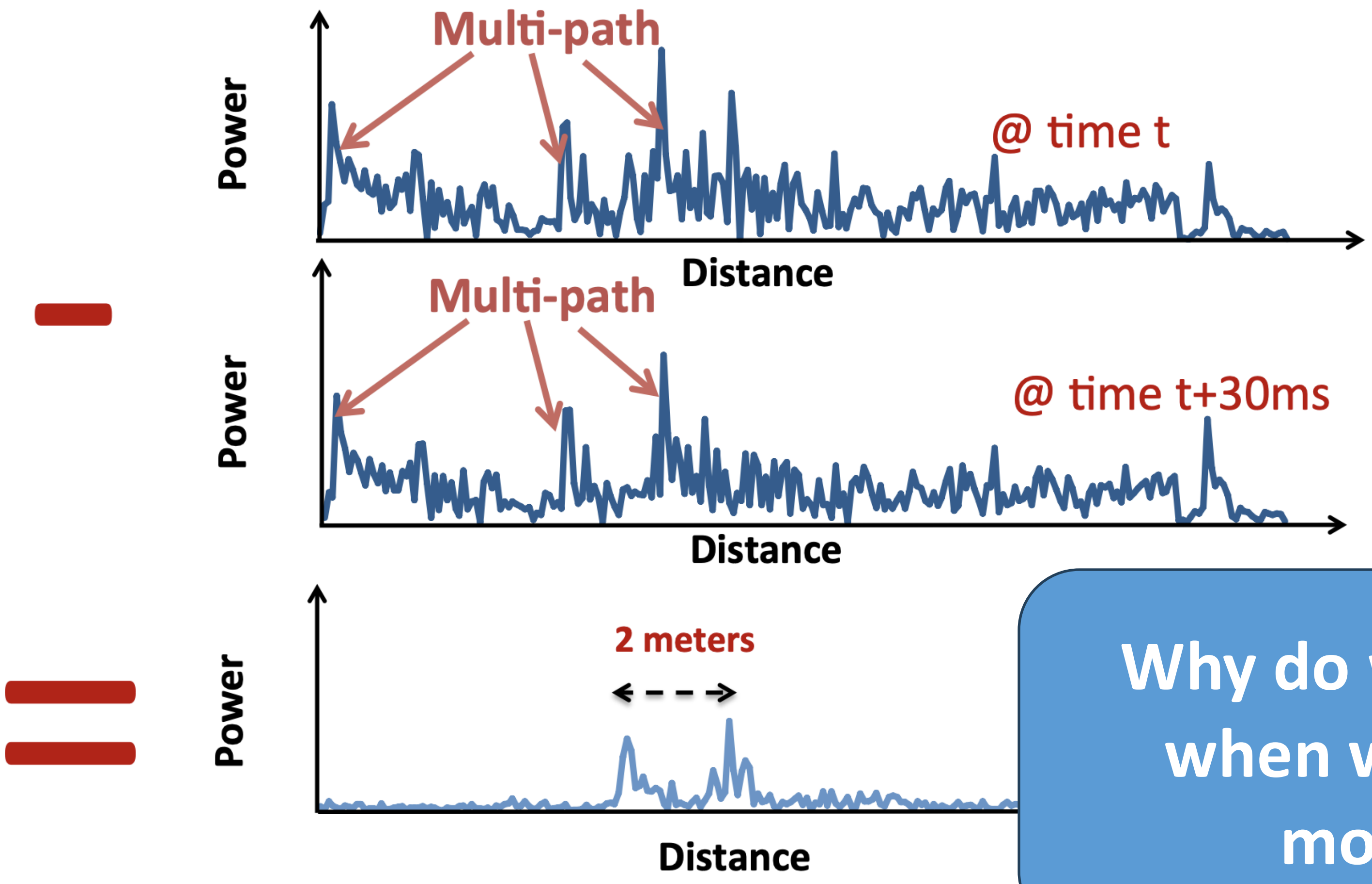
Handling Multipaths

- Static objects do not move – have the peak at fixed positions
  - Eliminate by subtracting consecutive measurements



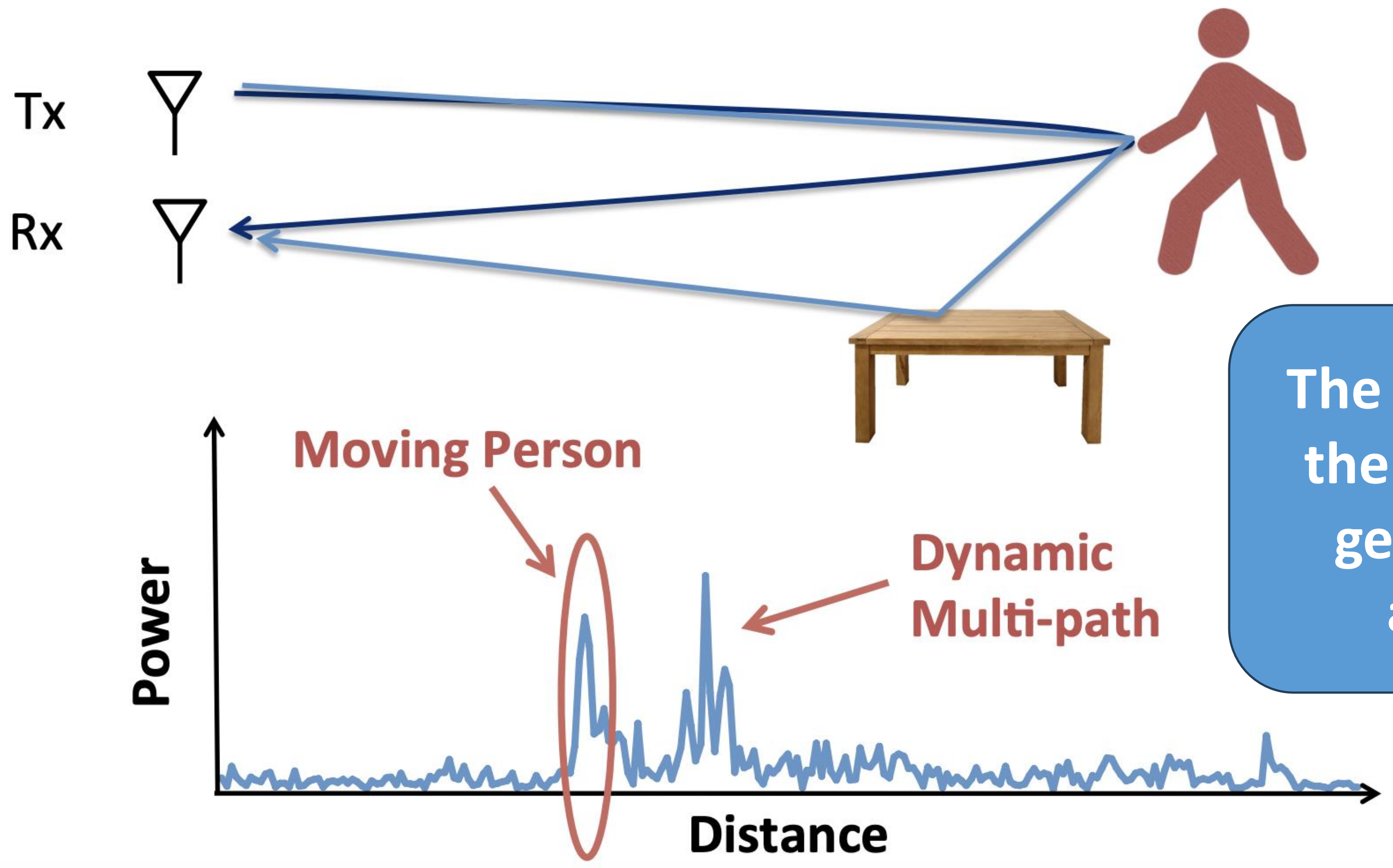
Handling Multipaths

- Static objects do not move – have the peak at fixed positions
  - Eliminate by subtracting consecutive measurements



Why do we see two peaks, when we have only one moving person?

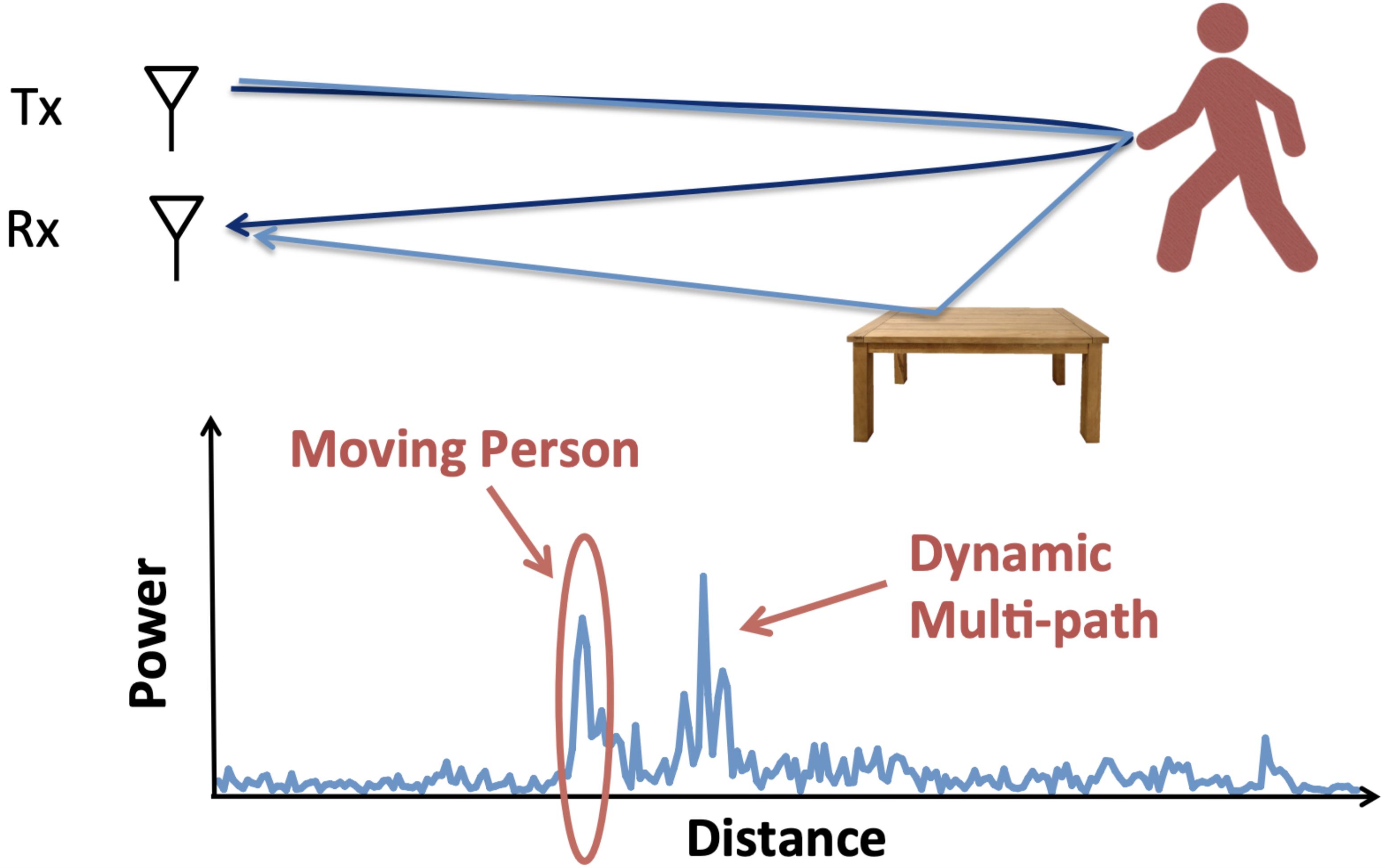
Dynamic Multipath



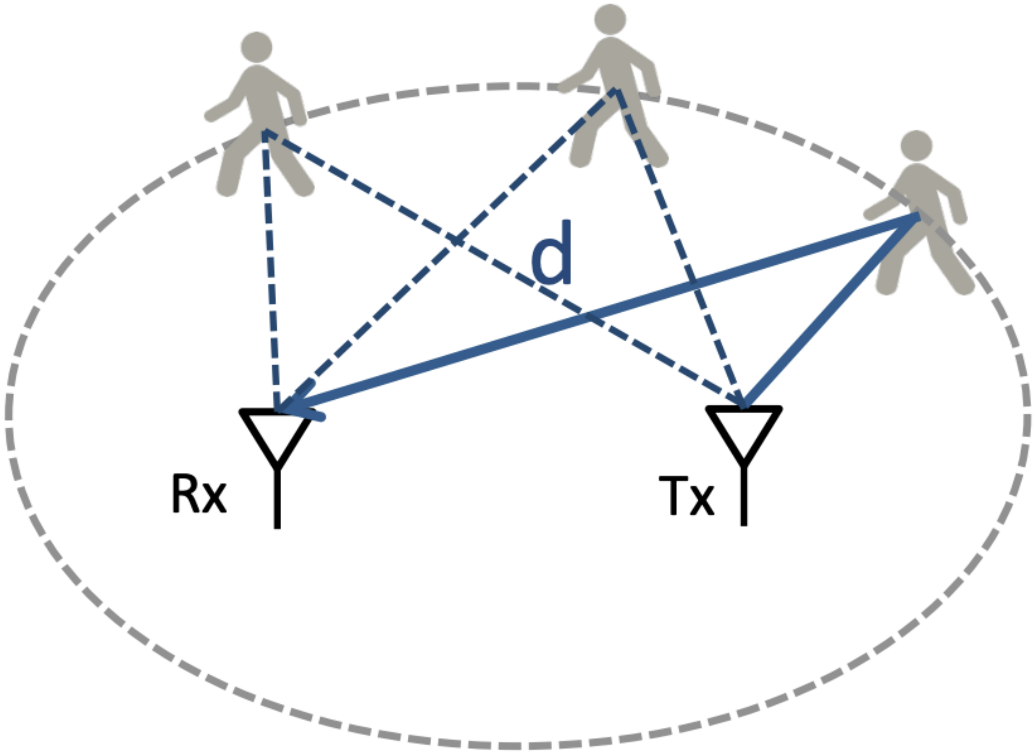
The reflected signal from the moving person may get reflected through another reflector

Handling Dynamic Multipath

- The direct reflection arrives before the dynamic multipath

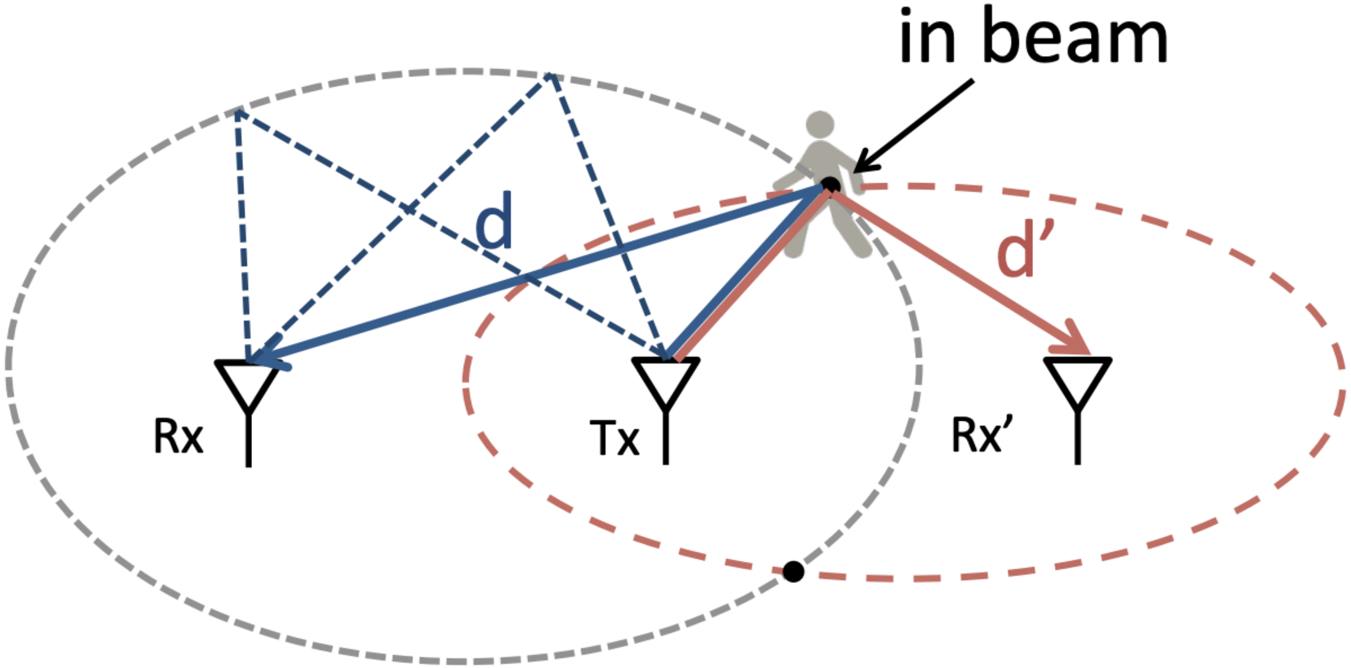


From Distance to Localization



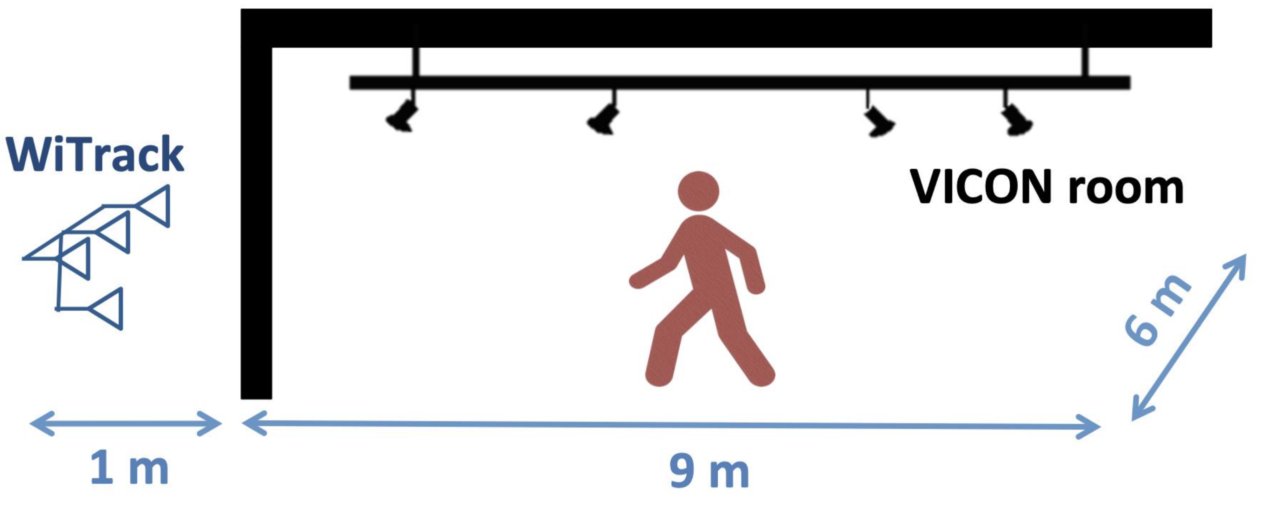
- Person can be anywhere in the ellipse whose foci are (Tx,Rx)
  - **One ellipse is not sufficient for localize**

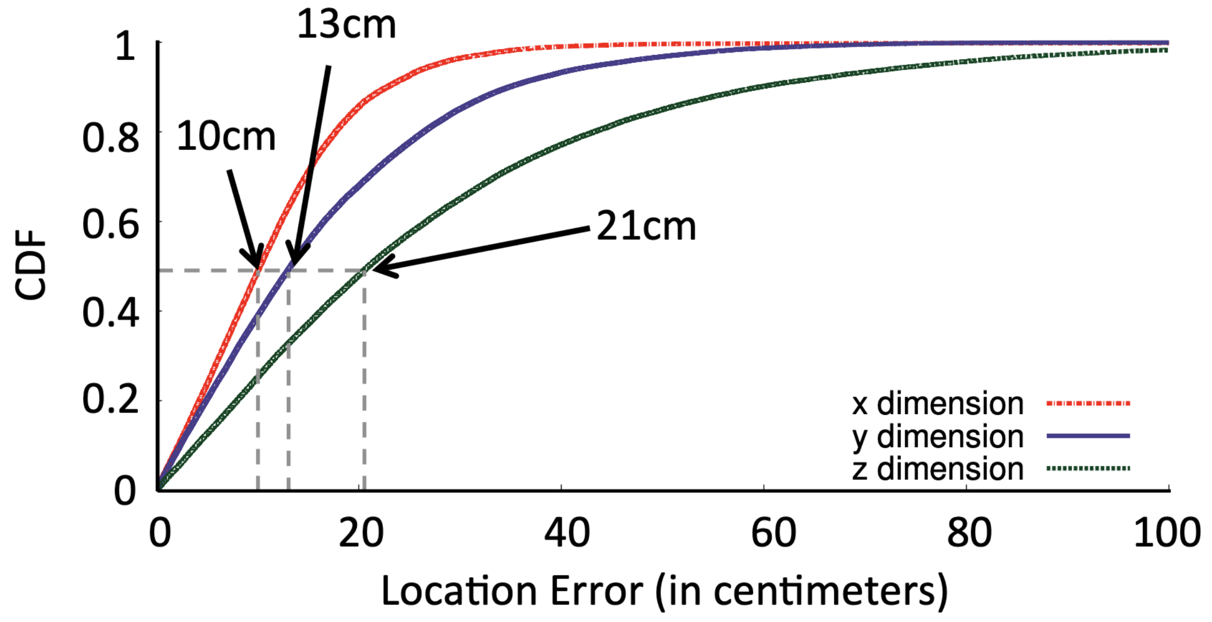
From Distance to Localization



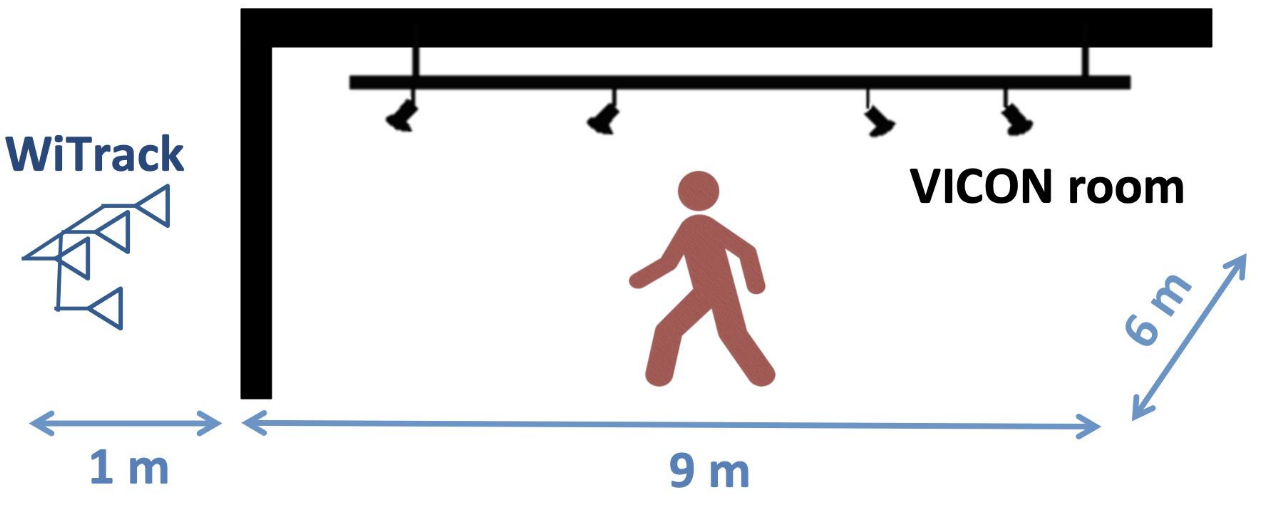
- Use multiple Rx to localize
  - WiTrack uses directional antennas; so only one point is in-beam
  - Use 3 Rx antennas for 3D localization

Performance

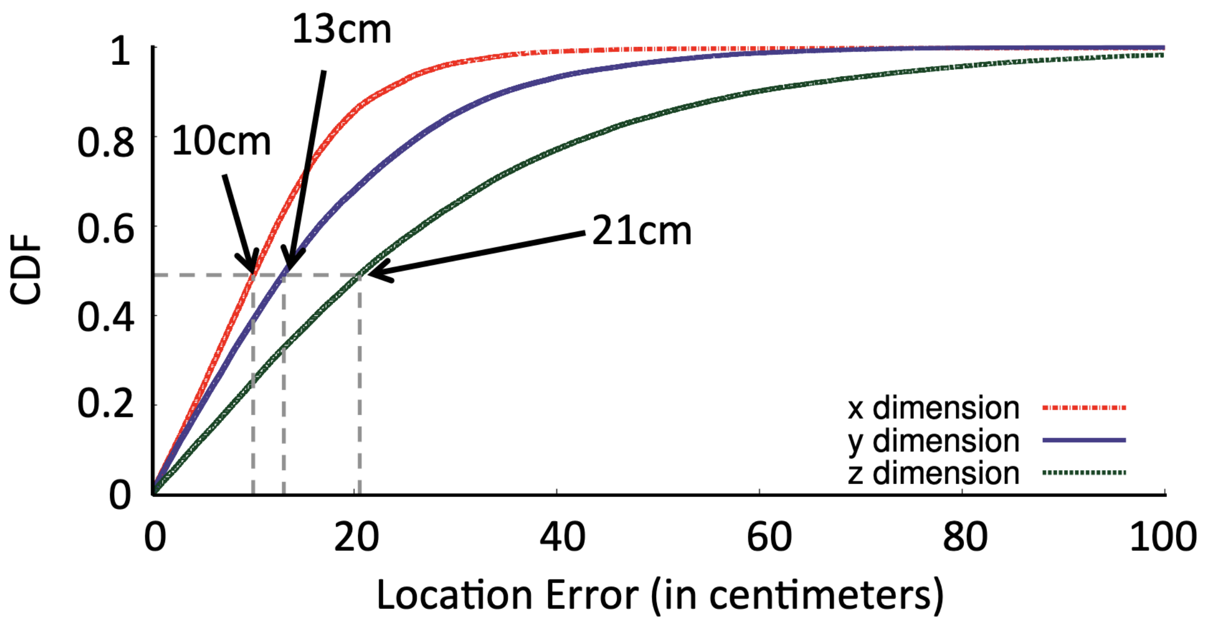




Performance

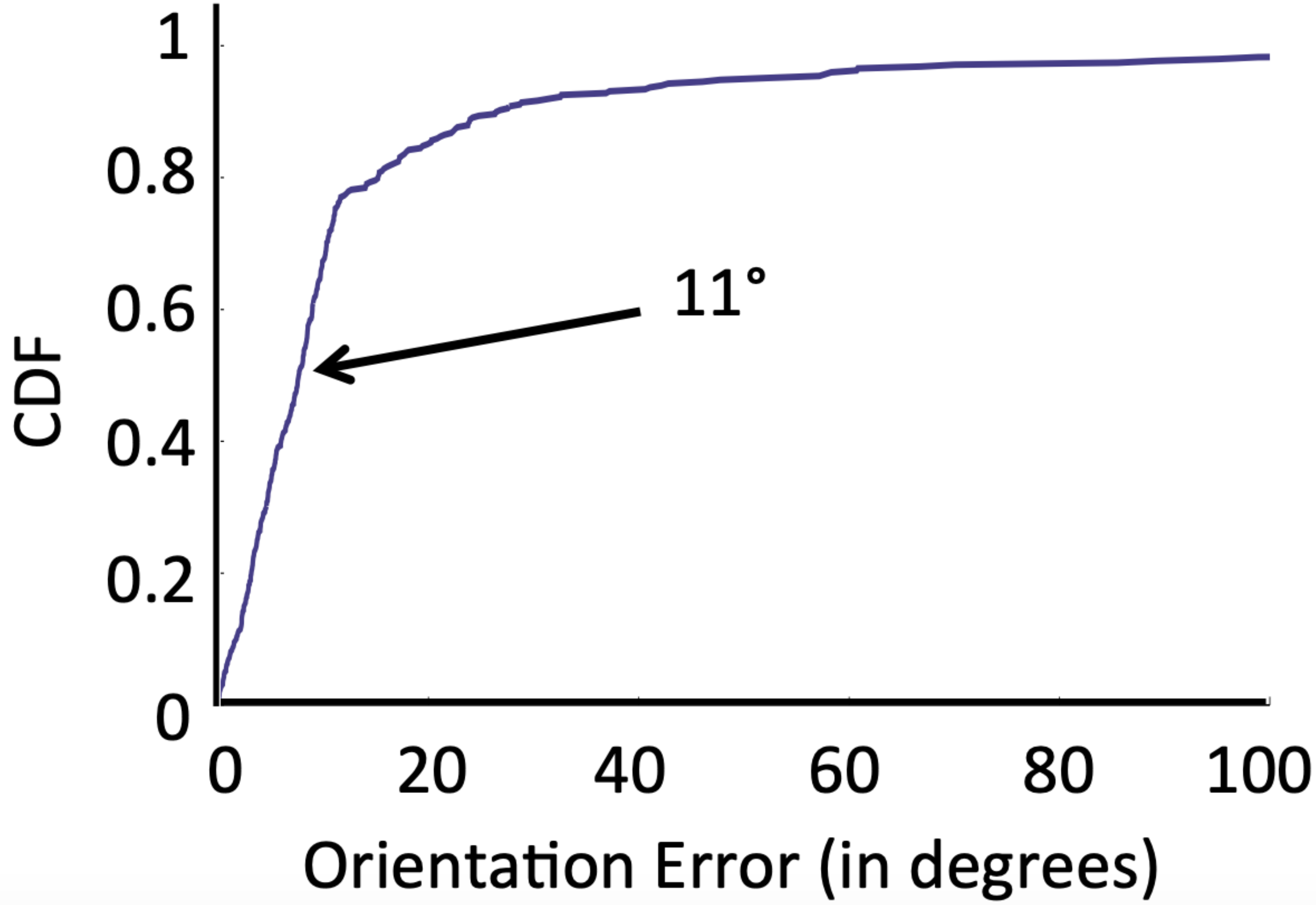


Centimeter-scale localization without requiring the user to carry a wireless device



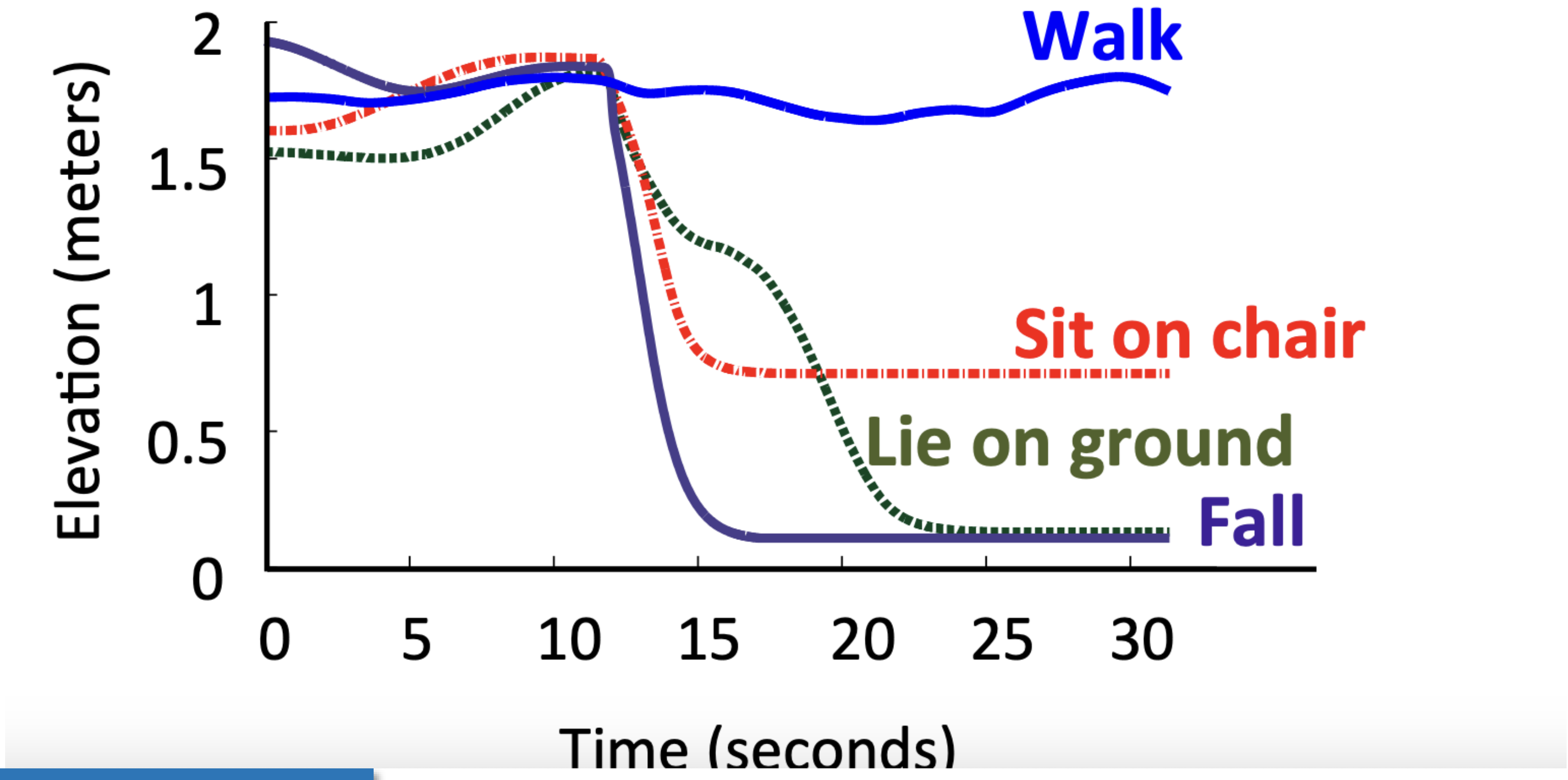
Performance

- Person points in a random direction



Performance

- Elderly monitoring to detect fall
    - Person simulates different activities



Summary

- The wireless signal is capable of monitoring the user movement patterns without requiring the user to carry any additional devices
  - Needs precise signal processing to estimate the reflection patterns

- Limitations of WiTrack
  - Can only detect a moving person
  - Cannot concurrently track multiple peoples

Passive Human Tracking with WiFi

Widar2.0: Passive Human Tracking with a Single Wi-Fi Link

Kun Qian[1], Chenshu Wu[2], Yi Zhang[1], Guidong Zhang[3], Zheng Yang[1], Yunhao Liu[1,4]
[1]Tsinghua University
[2]University of Maryland, College Park
[3]University of Science and Technology of China
[4]Michigan State University
{qiank10,wucs32,zhangyithss,hmilyyz,yunhaoliu}@gmail.com,zgd@mail.ustc.edu.cn

ABSTRACT

This paper presents Widar2.0, the first WiFi-based system that enables passive human localization and tracking using a single link on commodity off-the-shelf devices. Previous works based on either specialized or commercial hardware all require multiple links, preventing their wide adoption in scenarios like homes where typically only one single AP is installed. The key insight underlying Widar2.0 to circumvent the use of multiple links is to leverage multi-dimensional signal parameters from one single link. To this end, we build a unified model accounting for Angle-of-Arrival, Time-of-Flight, and Doppler shifts together and devise an efficient algorithm for their joint estimation. We then design a pipeline to translate the erroneous raw parameters into precise locations, which first finds parameters corresponding to the reflections of interests, then refines range estimates, and ultimately outputs target locations. Our implementation and evaluation on commodity WiFi devices demonstrate that Widar2.0 achieves better or comparable performance to state-of-the-art localization systems, which either use specialized hardwares or require 2 to 40 Wi-Fi links.

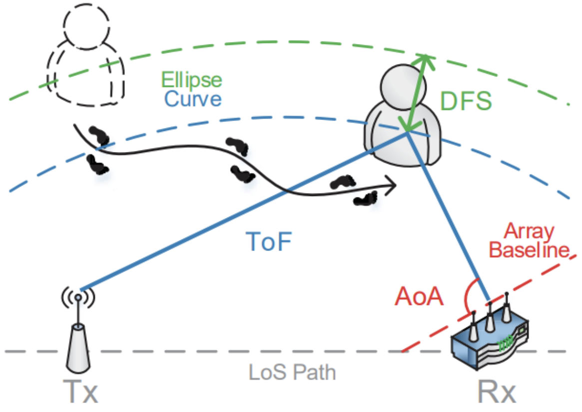


Figure 1: High-level design of Widar2.0. One receiver (e.g. laptop) overhears packet transmission from an AP and calculates ToF, AoA and DFS of the reflection path, for location estimate.

their applications in important scenarios like elderly care, security

ACM MobiSys 2018

Slides are based on the presentation by original authors

Motivation

- Passive localization helps in various applications

Smart Home

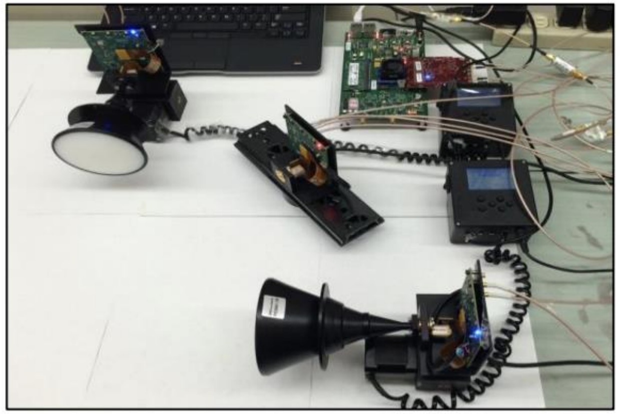
Health Monitoring

Intruder Detection

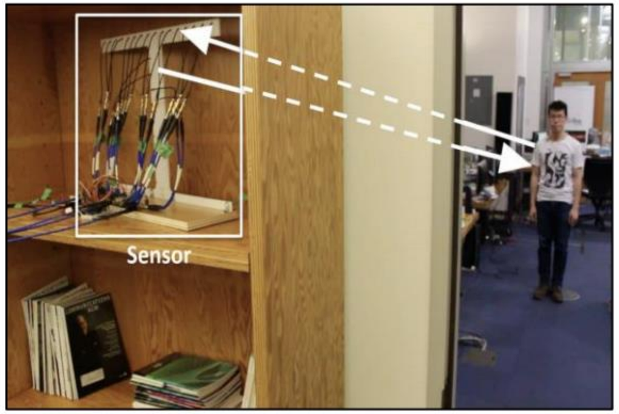
- RF vs Cameras
  - Less privacy concerns
  - Larger suveillance areas
  - More ubiquitous deployment

Motivation

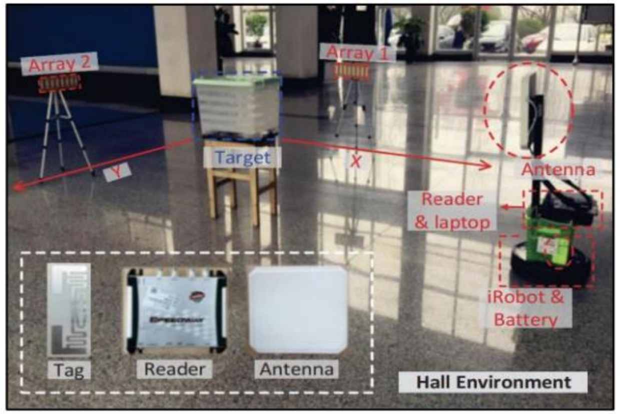
- RF-based tracking thrives with prevail RF devices



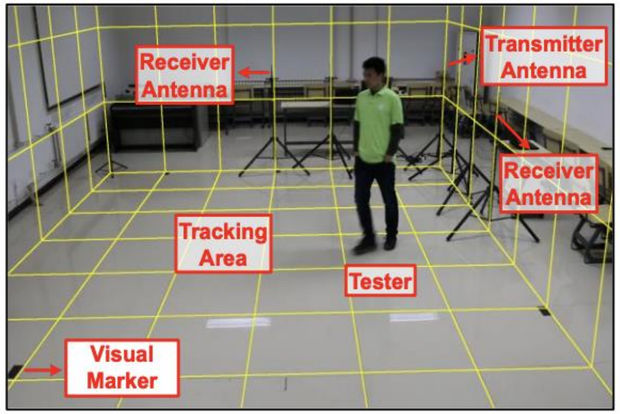
mmWave



FMCW Radar



RFID



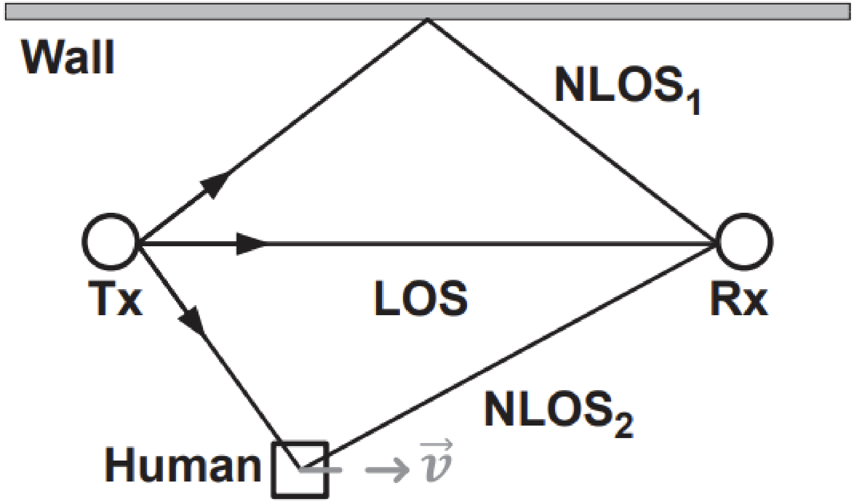
Wi-Fi

Widar: Exploring Doppler frequency shift for tracking

- CSI:

$$H(f,t) = \sum_{k=1}^{K} \alpha_k(t) e^{-j2\pi f \tau_k(t)},$$

- Movement introduces Doppler effect: changes the frequency of the observed at the Rx



Frequency shift of the signal reflected from human body:

$$f_D(t) = -\frac{1}{\lambda}\frac{\mathrm{d}}{\mathrm{d}t}d(t) = -f\frac{\mathrm{d}}{\mathrm{d}t}\tau(t), \quad (2)$$

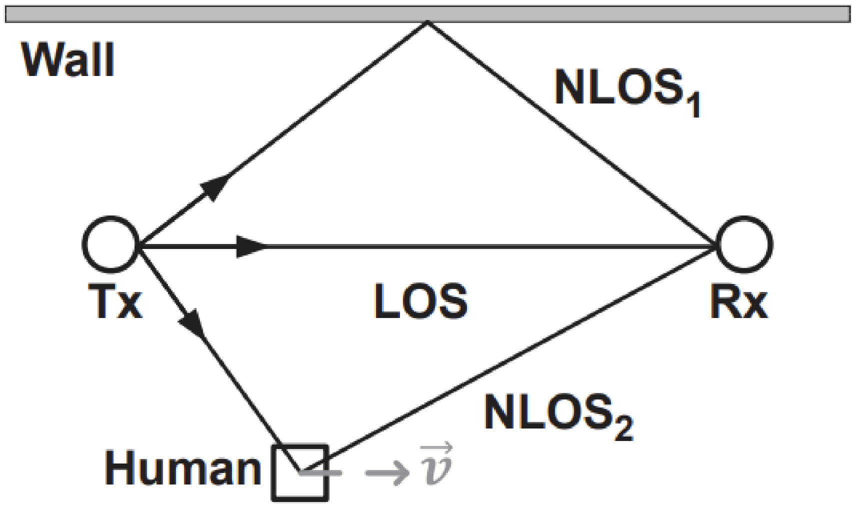
where λ, f, $\tau(t)$ are the wave-length, carrier frequency and time of flight of the signal, and $d(t)$ is the length of the path NLOS_2, and the PLCR r is mathematically defined as $r \triangleq \frac{\mathrm{d}}{\mathrm{d}t}d(t)$.

Widar: Exploring Doppler frequency shift for tracking

- CSI:

$$H(f,t) = \sum_{k=1}^{K} \alpha_k(t) e^{-j2\pi f \tau_k(t)},$$

- Movement introduces Doppler effect: changes the frequency of the observed at the Rx



CSI can be represented as superimposition of path responses modulated by Doppler frequency shift arising from moving reflectors on each path:

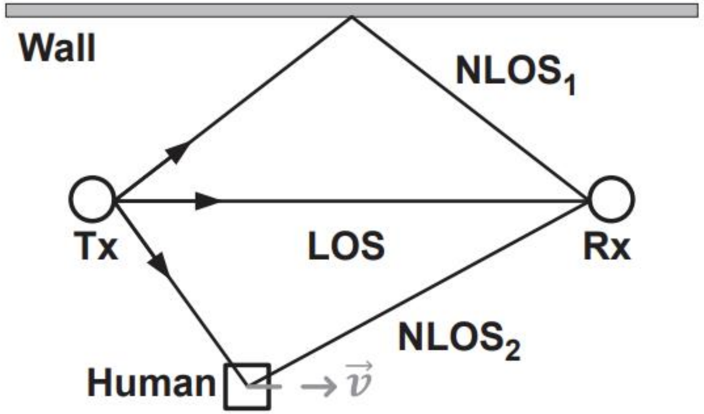
$$H(f,t) = \left(H_s(f) + \sum_{k \in P_d} \alpha_k(t) e^{j2\pi \int_{-\infty}^{t} f_{D_k}(u)\mathrm{d}u}\right) e^{-j\kappa(f,t)},$$

Static Path

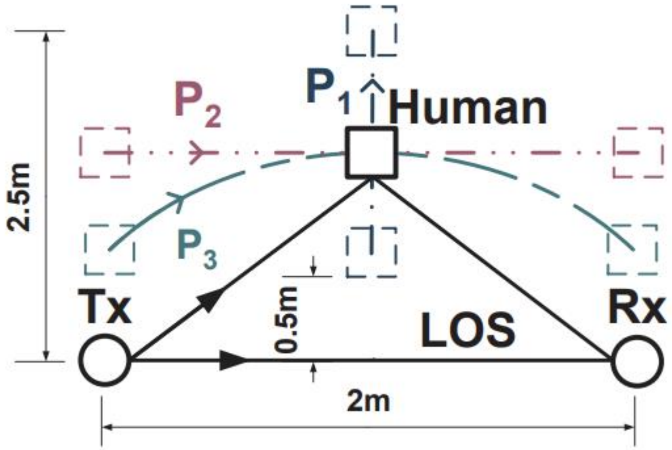
Dynamic Path

Widar: Exploring Doppler frequency shift for tracking

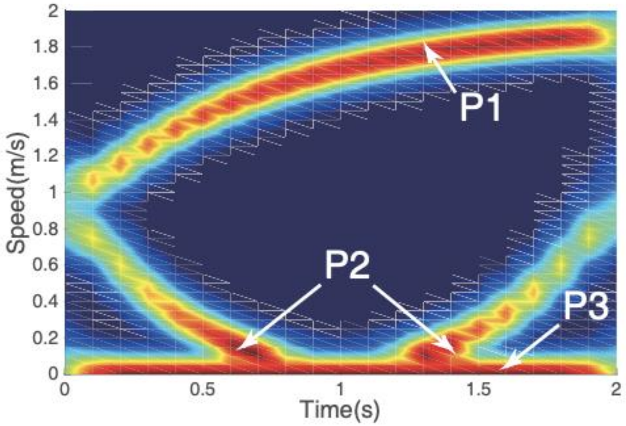
- Explore Doppler frequency shifts (DFS) over multiple links to compute Path Length Change Rate (PLCR)



(a) Multipath propagation.



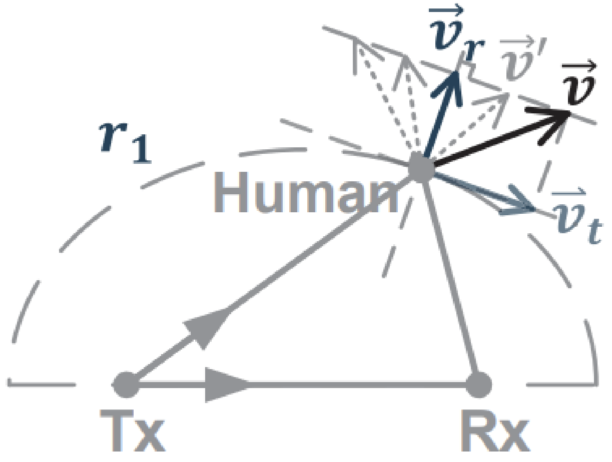
(b) Human walking paths P_1, P_2, P_3



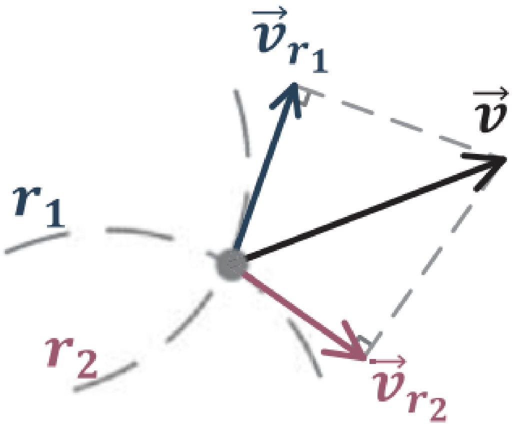
(c) PLCRs calculated from the three scenario in (b)

Widar: Exploring Doppler frequency shift for tracking

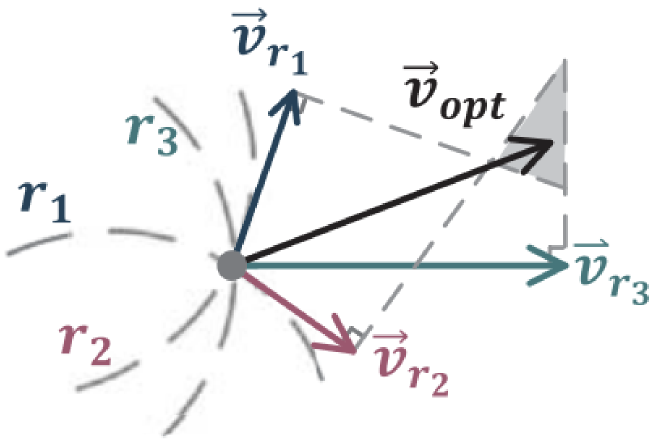
- Multiple links give more precise path information



One Link



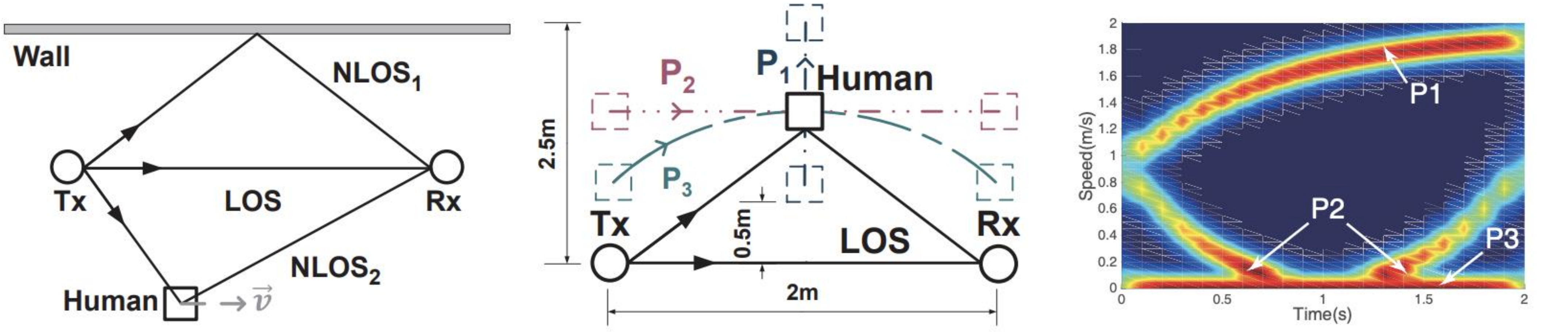
Two Links



Three Links

Widar: Exploring Doppler frequency shift for tracking

- Explore Doppler frequency shifts (DFS) over multiple links to compute Path Length Change Rate (PLCR)



(a) Multipath propagation. (b) Human walking paths P_1, P_2, P_3 (c) PLCRs calculated from the three scenario in (b)

- Widar requires
  - DFS from multiple links to compute velocity
  - Trial and error to reduce direction ambiguity
  - Costly search to spot the initial location

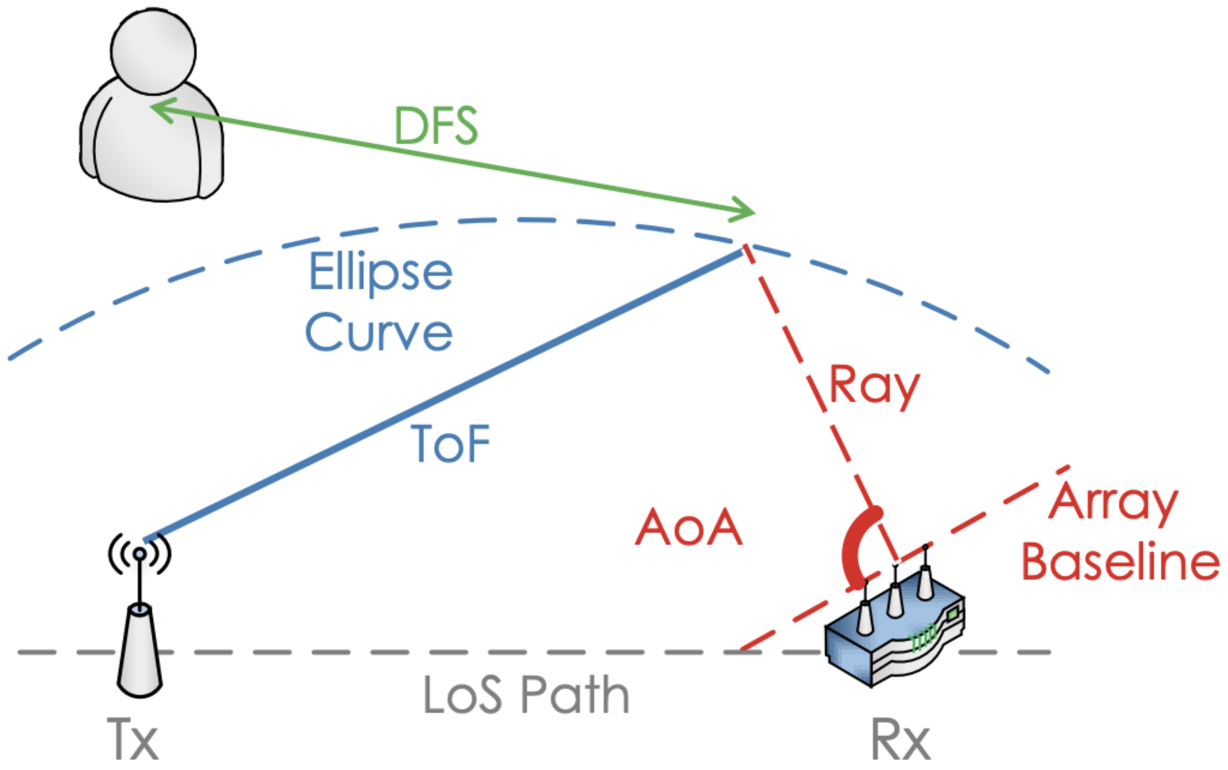
Other Related Works in the Literature

- Existing approaches need
  - Single Link but specialized hardware: **Less ubiquitous**
  - Commercial devices but multiple links: **Less practical**

WiTrack **WiDeo** **Widar** **Dynamic MUSIC** **LiFS**

Key Idea

- Can we achieve both ubiquity and practicality?
    - Using a single commercial WiFi link

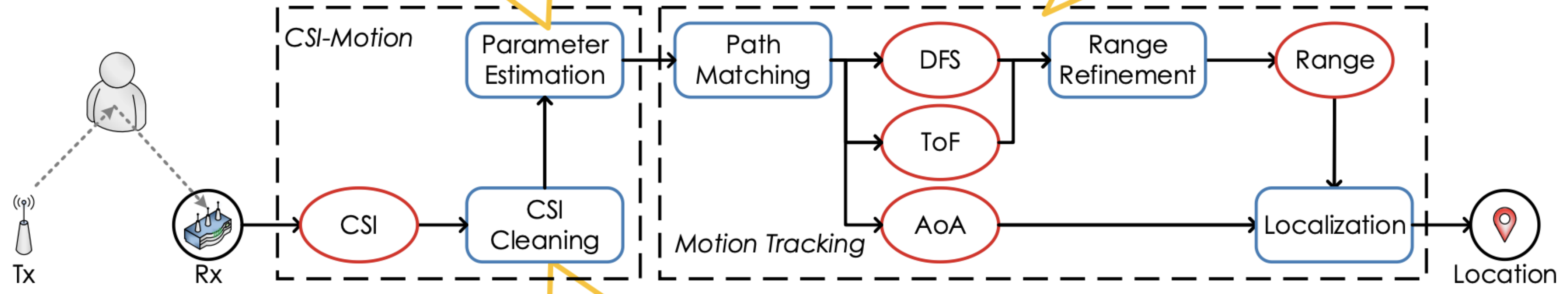


- Track multiple signal parameters instead of just one, and then combine the information from all
    - Angle of Arrival (AoA)
    - Time of Flight (ToF)
    - Doppler frequency shift (DFS)

Widar 2.0: Broad System Overview

C1: How to jointly estimate multipath parameters in CSI?

C3: How to derive locations from un-matched parameters?



C2: How to calibrate CSI phase noises?

CSI Model

- As discussed, due to multipath effect, we can model CSI as

$$H(t,f,\mathbf{s}) = \sum_{l=1}^{L} P_l(t,f,\mathbf{s}) + N(t,f,\mathbf{s}) = \sum_{l=1}^{L} \alpha_l(t,f,\mathbf{s}) e^{-j2\pi f \tau_l(t,f,\mathbf{s})} + N(t,f,s)$$

- Wi-Fi NICs measure channel discretely in time (packet), frequency (subcarrier) and space (sensor)

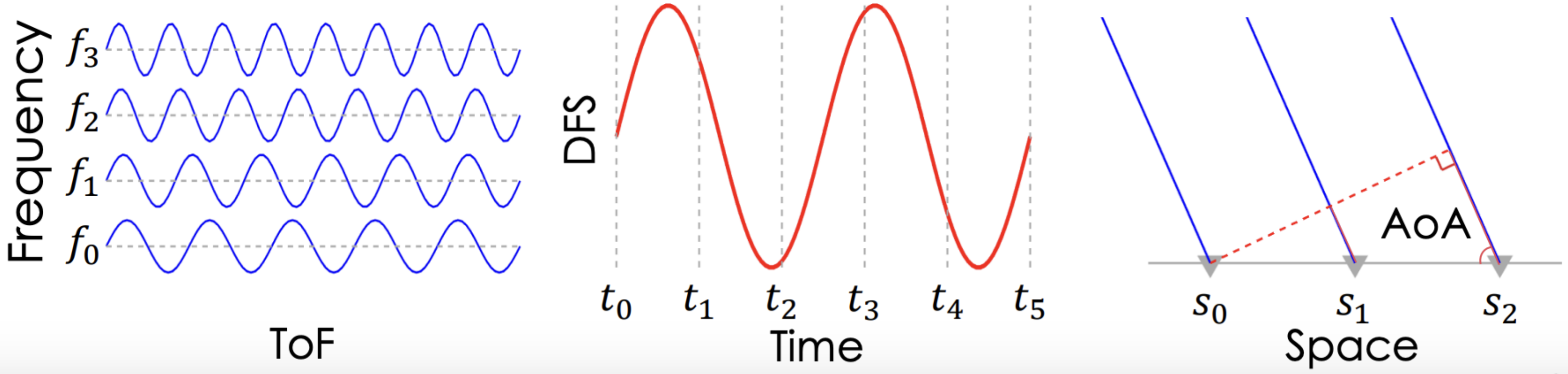
CSI Model

- As discussed, due to multipath effect, we can model CSI as

$$H(t,f,\boldsymbol{s}) = \sum_{l=1}^{L} P_l(t,f,\boldsymbol{s}) + N(t,f,\boldsymbol{s}) = \sum_{l=1}^{L} \alpha_l(t,f,\boldsymbol{s}) e^{-j2\pi f \tau_l(t,f,\boldsymbol{s})} + N(t,f,s)$$

- The delay of the l-th path $\tau_l\,(i,j,k)$ is a combination of ToF τ_l, DFS f_{Dl} and AoA $\boldsymbol{\phi}_l = (\cos\phi_l, \sin\phi_l)^T$

$$\Delta f_j \tau_l \quad + f_{D_l} \Delta t_i \quad -\frac{1}{\lambda} \Delta \boldsymbol{s_k} \cdot \boldsymbol{\phi}_l$$



Parameter Estimation

- Modeled as a maximal likelihood estimation (MLE)

The MLE of $\boldsymbol{\theta}_l = (\alpha_l, \tau_l, \phi_l, f_{D_l})$ for all paths, $\boldsymbol{\Theta} = (\boldsymbol{\theta}_l)_{l=1}^{L}$ is formulated as:

$$\Lambda(\Theta; H) = -\sum_{i,j,k} \left| H(i,j,k) - \sum_{l=1}^{L} P_l(i,j,k;\theta_l) \right|^2$$

- L is the number of multipath
  - L should be larger than the number of principal multipaths
  - Widar 2.0 considers L=5

SAGE Algorithm

- Space Alternating Generalized Expectation (SAGE) Maximization: A general version of EM algorithm
  - Each iteration of the algorithm only re-estimates a subset of the components of Θ while keeping the estimations of the othercomponents fixed

- Estimation step: $$\hat{p}_l(i,j,k;\widehat{\Theta}') = P_l(i,j,k;\hat{\theta}_l') + \beta_l\left(H(i,j,k) - \sum_{l'=1}^{L} P_l(i,j,k;\hat{\theta}_{l'}')\right)$$

- Maximization step:

$$\hat{\tau}_l'' = argmax_\tau\{|z(\tau,\hat{\phi}_l',\hat{f}_{D_l}';\hat{p}_l(i,j,k;\widehat{\Theta}')|\}$$

$$\hat{\phi}_l'' = argmax_\phi\{|z(\hat{\tau}_l'',\phi,\hat{f}_{D_l}';\hat{p}_l(i,j,k;\widehat{\Theta}')|\}$$

$$\hat{f}_{D_l}'' = argmax_{f_D}\{|z(\hat{\tau}_l'',\hat{\phi}_l'',f_D;\hat{p}_l(i,j,k;\widehat{\Theta}')|\}$$

$$\hat{\alpha}_l'' = \frac{z(\hat{\tau}_l'',\hat{\phi}_l'',\hat{f}_{D_l}'';\hat{p}_l(i,j,k;\widehat{\Theta}')}{TFA}$$

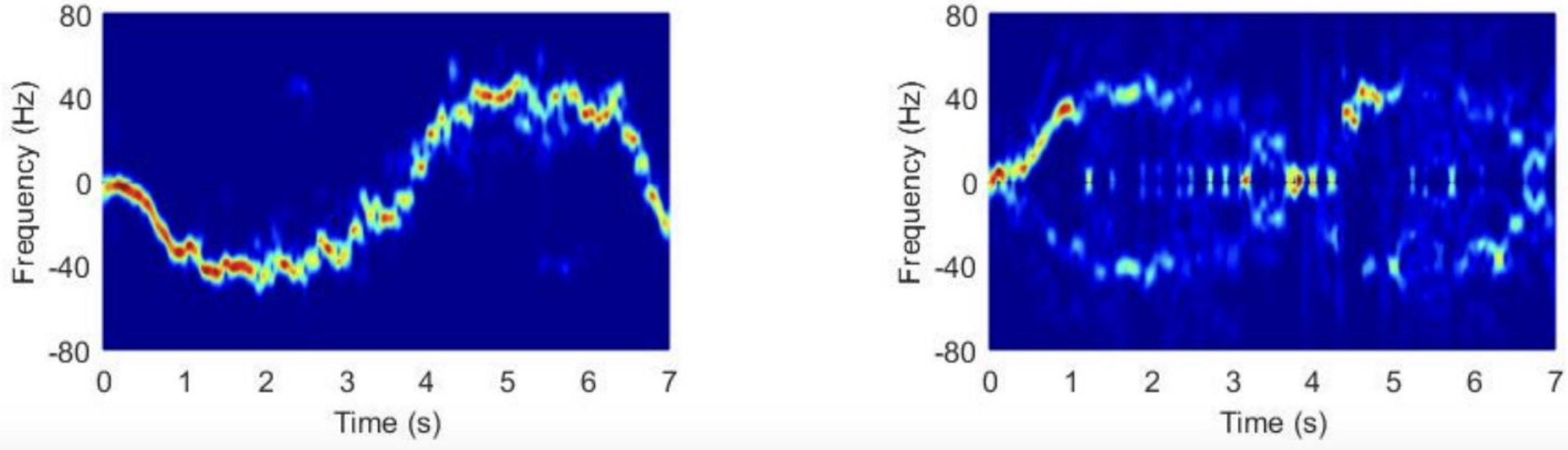
$$z(\tau,\phi,f_D;P_l) = \sum_{i,j,k} e^{2\pi\Delta f_j \tau_l} e^{2\pi f_c \Delta s_k \cdot \phi_l} e^{-2\pi f_{D_l} \Delta t_i} P_l(i,j,k)$$

CSI Cleaning

- CSI contains not only channel response, but also various unknown phase noises (timing offset and carrier frequency offset)

$$\widetilde{H}(i,j,k) = H(i,j,k)e^{2\pi\left(\Delta f_j \epsilon_{t_i} + \Delta t_i \epsilon_f\right)}$$

- Linear Regression calibration does not work (assumption: Phase noises are linear across subcarriers)
  - Weak reflection from human body



CSI Cleaning

- Code idea used in Widar 2.0
  - CSI phase noises caused by TO and CFO only vary in time and frequency, but not space.
  - All sensors of the same NIC experience the same unknown phase noises at the same time

- Selects a sensor as the reference sensor (e.g. k_0-th sensor), calculate the conjugate multiplications C(m), between CSIs of each sensor and the reference sensor

CSI Cleaning

- Use conjugate multiplication between each antenna and chosen reference antenna

$$C(i,j,k) = \widetilde{H}(i,j,k) * \widetilde{H}^*(i,j,k_0)$$

- By classifying multipath into static signals P_s (f_D = 0) and dynamic signals P_d ($f_D \neq 0$) , we have

$$\begin{aligned} C(i,j,k) = & \sum_{n_1,n_2 \in P_s} P_{n_1}(i,j,k) P_{n_2}^*(i,j,k_0) \\ & + \sum_{l \in P_d, n \in P_s} \underbrace{P_l(i,j,k) P_n^*(i,j,k_0)}_{\text{Target term}} + P_n(i,j,k) P_l^*(i,j,k_0) \\ & + \sum_{l_1,l_2 \in P_d} P_{l_1}(i,j,k) P_{l_2}^*(i,j,k_0) \end{aligned}$$

CSI Cleaning

- Use conjugate multiplication between each antenna and chosen reference antenna

$$C(i,j,k) = \widetilde{H}(i,j,k) * \widetilde{H}^*(i,j,k_0)$$

- By classifying multipath into static signals P_s (f_D = 0) and dynamic signals P_d ($f_D \neq 0$) , we have

$$C(i,j,k) = \sum_{n_1,n_2 \in P_s} P_{n_1}(i,j,k) P_{n_2}^*(i,j,k_0)$$

Remove through high pass filter, as static signals are constant

$$+ \sum_{l \in P_d, n \in P_s} \underbrace{P_l(i,j,k) P_n^*(i,j,k_0)}_{\text{Target term}} + P_n(i,j,k) P_l^*(i,j,k_0)$$

$$+ \sum_{l_1,l_2 \in P_d} P_{l_1}(i,j,k) P_{l_2}^*(i,j,k_0)$$

Much weaker than the first two, as static signals are much stronger

CSI Cleaning

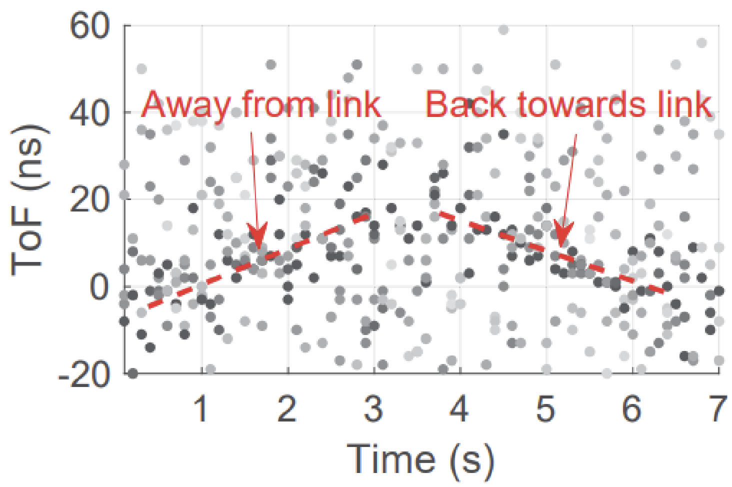
- The target term preserves the CSI phase

$$P_l(\boldsymbol{m})P_n^*(\boldsymbol{m}_0) = \alpha_l \alpha_n^* e^{-2\pi \Delta f_j(\tau_l - \tau_n) - 2\pi f_c \Delta \boldsymbol{s}_k \cdot \boldsymbol{\phi}_l + 2\pi f_{D_l} \Delta t_i}$$

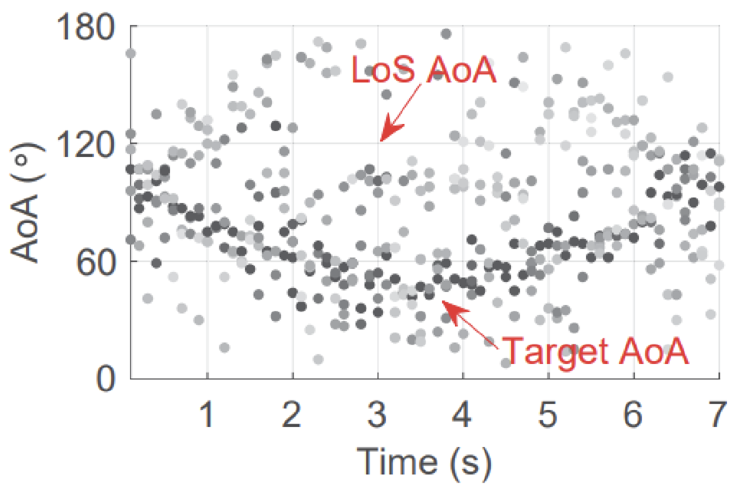
$$P_n(\boldsymbol{m})P_l^*(\boldsymbol{m}_0) = \alpha_n \alpha_l^* e^{-2\pi \Delta f_j(\tau_n - \tau_l) - 2\pi f_c \Delta \boldsymbol{s}_k \cdot \boldsymbol{\phi}_n - 2\pi f_{D_l} \Delta t_i}$$

Path Matching

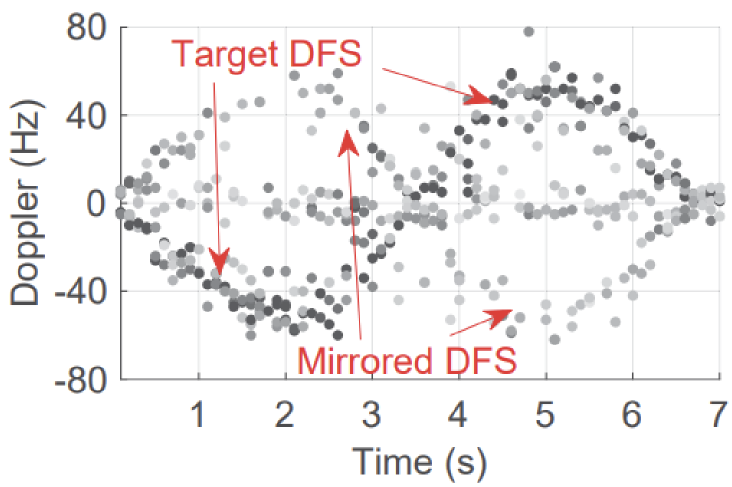
- Multipath parameters are cluttered together



(a) ToF



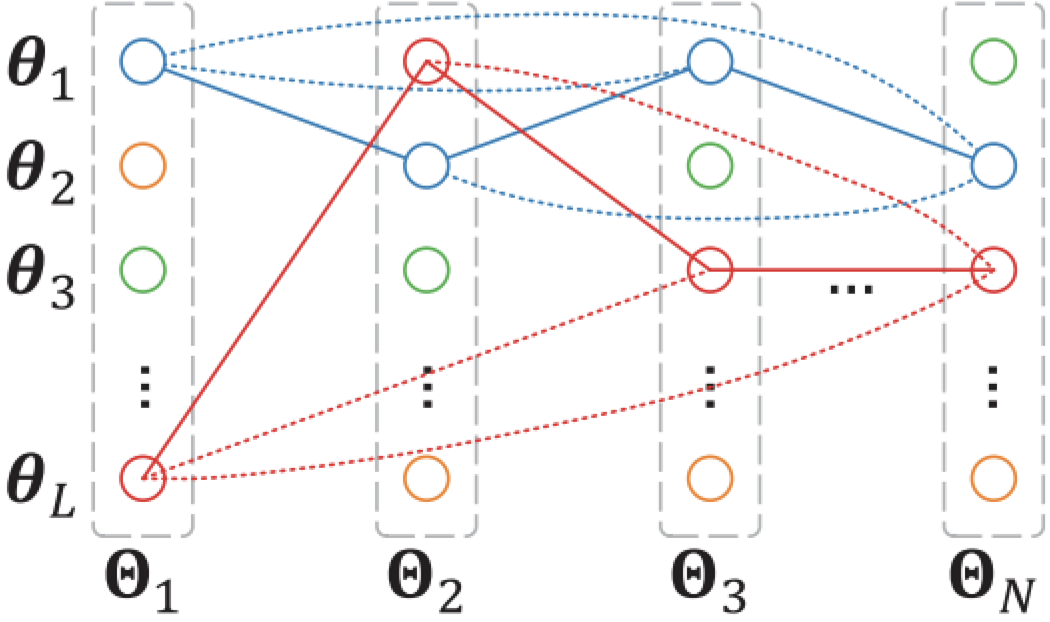
(b) AoA



(c) DFS

Path Matching

- Graph-based Path Matching (GPM)
  - Model successive parameters as the weights of a N-partite graph for N CSI segments
  - The joint estimate of the parameter values does not change significantly across successive CSI measurements, so the total path weight would be minimum



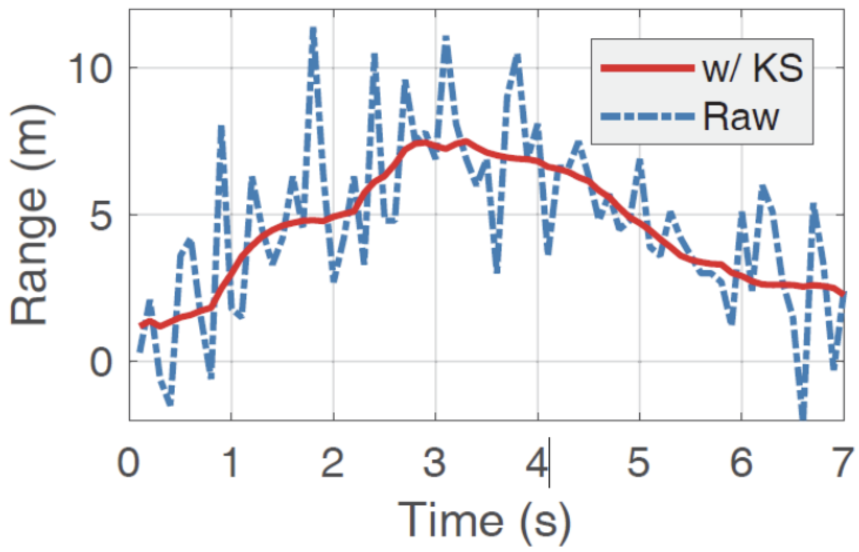
edge wight is modeled as the distance between parameters

$$w_{i_1 j_1}^{i_2 j_2} = w_{i_2 j_2}^{i_1 j_1} = ||\boldsymbol{c}^T(\boldsymbol{\theta}_{i_1 j_1} - \boldsymbol{\theta}_{i_2 j_2})||$$

Range Refinement

- Range estimation with ToF or DFS
  - ToF -> coarse estimate of absolute range
  - DFS -> fine estimate of change rate of range

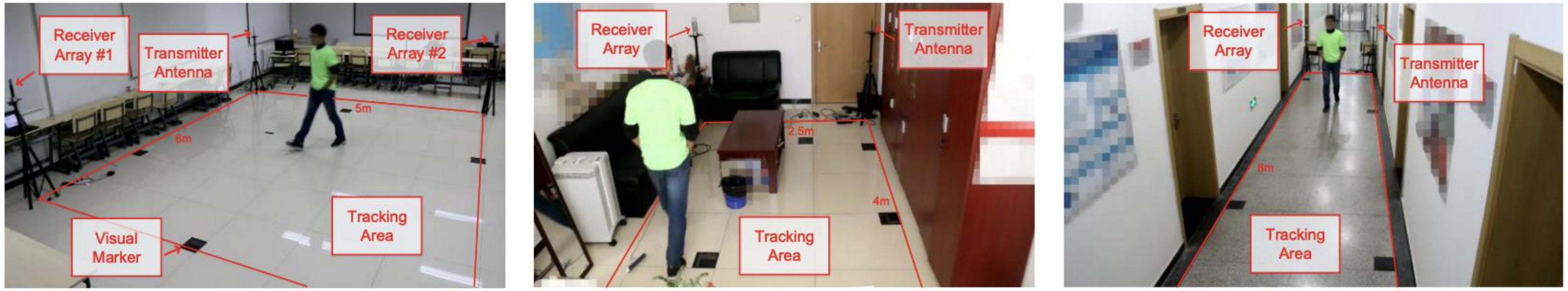
- Use Kalman filter to refine range with both ToF and DFS



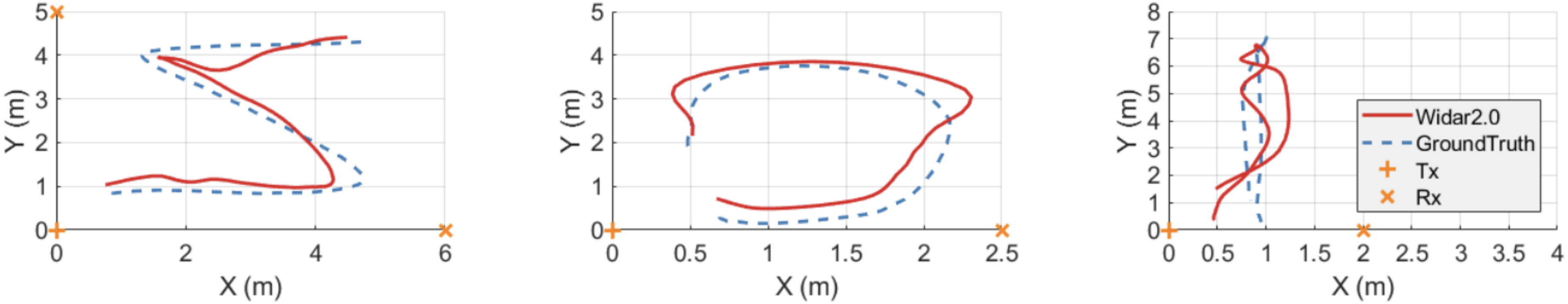
Example of range refinement.

Experiment

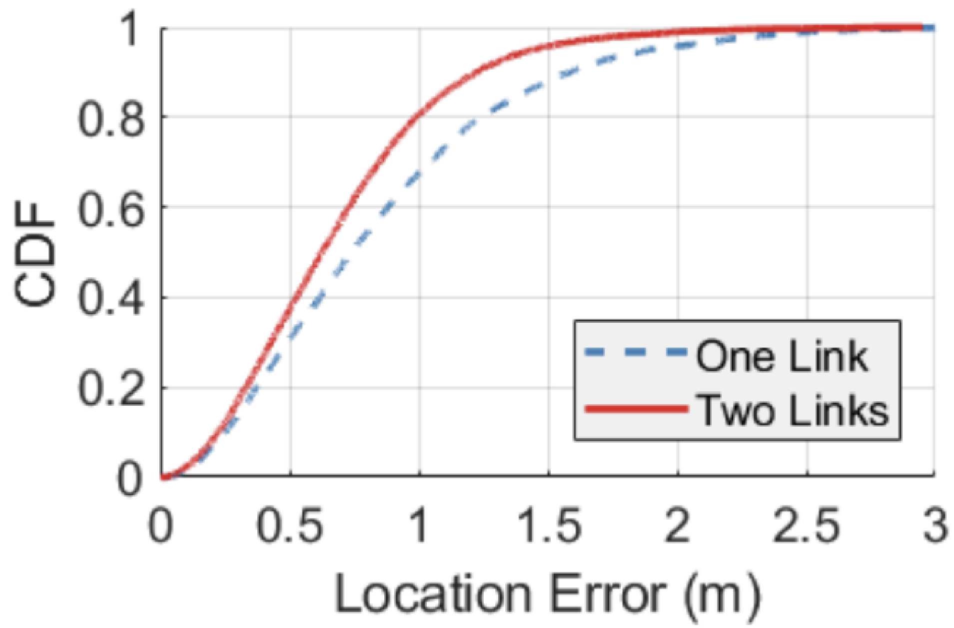
- Implementation
  - Thinkpad laptops with Intel 5300 NIC
- Setup
  - 3 scenarios: classroom, corridor, office



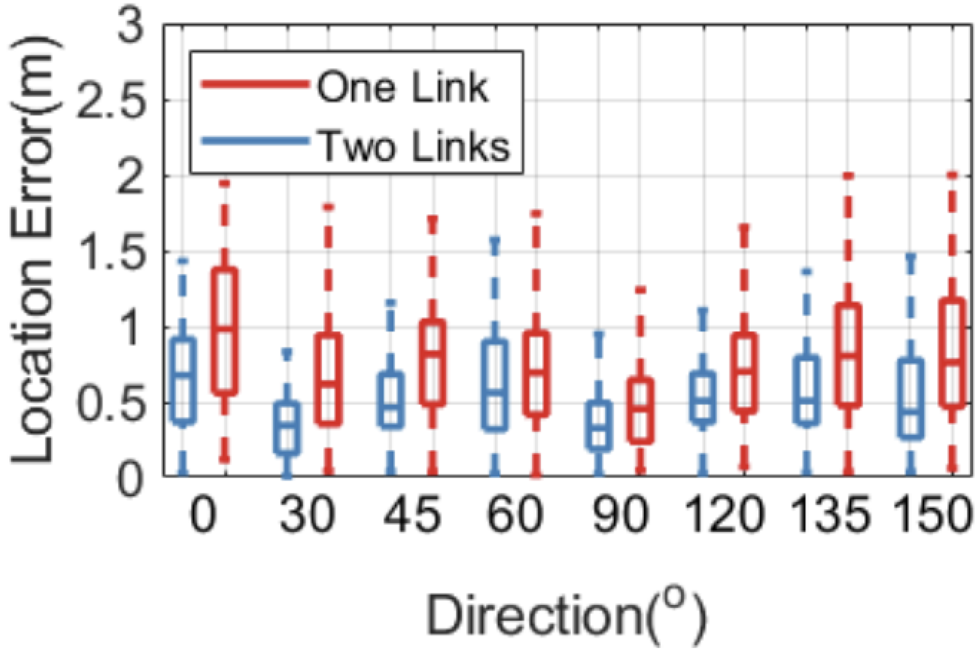
- Tracking samples



Performance



Overall Localization Accuracy



Walking Direction

Summary

- Robust multi-parameter RF tracking
  - Over a single link
  - A unified model of ToF, AoA and DFS
  - CSI calibration for weak reflection path
  - Robust parameter matching and refinement for localization

- Decimeter-level passive tracking system
  - Median location error of 7.5dm with one single link

Happy Learning!

Some resources related to this topic