Basic idea:
- Transmit beeps from A, then from B.
- Receive A's beep at A's mic & B's mic. and vice versa.
- Compute range from JUST these measurements.
- Robust to clock sync., OS delays, clock drifts.

Technique:
(I) \( d_{AB} = (t'_B - t'_A) \cdot c \) where \( c \) is 340 m/s.

But \( t_B \) and \( t_A \) are not clk synchronized, so this will not work.

(II) How about:

\[
d_{AB} = \frac{1}{2} ((t'_B - t'_A) + (t''_A - t''_B)) \cdot c
\]

\[
= \frac{1}{2} ((t''_A - t'_A) - (t''_B - t'_B)) \cdot c
\]

A's local clk \quad B's local clk

So this works well even w/o clk synchronization.

But what if OS delays mean that the timestamp is not when the signal was actually received.

(III)

\[
d_{AB} = \frac{c}{2} \left\{ (t''_A - t'_A) - (t''_B - t'_B) + (t''_B - t''_B) + (t''_B - t''_B) \right\}
\]

\[
= \frac{c}{2} \left\{ (t''_A - t'_A) + (t''_B - t''_B) + (t''_B - t''_B) - (t''_B - t''_B) \right\}
\]

\[
= \frac{c}{2} \left\{ d_{AA} + d_{BB} + \Delta_A + \Delta_B \right\}
\]
\[
\delta = \frac{c}{2} \left( \sum d_{AA} + d_{BB} + \Delta_A + \Delta_B \right)^2
\]

time gap between receiving self signal \& B's signal at Mic A.

Same as \( \Delta_A \) but at Mic B.

NOTE:

\( \sum d_{AA} \& d_{BB} \) can be measured a priori, and one time is adequate.

(3) Consider \( \Delta_A \):

- It's the same microphone, so then time can be measured in number of samples as opposed to relying on OS time stamps.

(3) How to measure \( \Delta_A \)?

- Correlate for A's own signal and assume correlation spikes at sample \# X
- Again correlate for B's signal and say spike happens at sample \# Y
- Compute \( (Y - X) \) in terms of \# of samples as measured by ADC.
ADC sampling rate known, hence

\[(Y - X) \cdot \frac{1}{f_s} = \text{time difference } \Delta_A\]

Here \(f_s\) is sampling rate of ADC.

Note: Both \(\Delta_A\) and \(\Delta_B\) are free of OS delays. The values get computed at the PHY layer itself so long as the mic. is ON continuously for the signal exchange.

Protocol:

1. Design the “beep” to have good auto correlation.

2. When detecting \(X\) and \(Y\), take the first peak in correlation since that line of sight (LOS) path. This gives robustness to multipath.

3. Compute and exchange \(\Delta_A\) and \(\Delta_B\). Also exchange \(\Delta_{AA}\) and \(\Delta_{BB}\).

4. Now both A and B computes \(\Delta_{AB}\).
Questions?
Goal: Track a smartphone's 3D motion in air using sound from multiple speakers.

Application: TV remote control.

Technique Overview:
1. Assume initial location of phone
2. Assume distance $D$ between speakers
3. Let two speakers send freq. tones $f_i$ and $f_j$
4. Compute doppler at phone, infer velocity, and track 2D motion.

Details:
1. Doppler Equation: $V = \frac{F^s}{F} \cdot C \Rightarrow F^s = \frac{V \cdot F}{C}$

where $V$ is velocity of the device,
$F^s$ is the shift in freq., $F$ is freq. of sound,
$C$ is vel. of sound in air.
• Send tones at $f_i$ and $f_j$ such that $f_i, f_j > 17$ kHz so they are inaudible.

• Sampling rate of signal is 44 kHz, which is standard for Android.

3) Take **44000 point FFT**, so you get freq. resolution at 1 Hz. Get spectrogram.

\[
\text{freq.} \quad \text{freq. shift} \quad \text{time}
\]

4) displacement = \(\int_{t_0}^{t_0+T} u \, dt = \int_{t_0}^{t_0+T} \frac{F_s}{F} \cdot c \, dt\)

\[
= \frac{c}{F} \int_{t_0}^{t_0+T} F_s \, dt = \frac{c}{F} \int_{t_0}^{t_0+T} \text{FFT} \text{(spectrogram)} \, dt
\]
Pick the one closer to the last estimate.

1. Improve Doppler accuracy.
   - Transmit bunch of tones from each speaker
   - Maximal ratio combining (MRC) to combine estimates from each tone
     - Note MRC is weighted average where weights are \( \frac{1}{\text{noise variance}} \)
     - Optimal when noise is Gaussian
   - Eliminate outliers for better MRC
   - Apply Kalman filter for smoothing.

2. Relaxing Assumptions:
   ① Distance between speakers.
   - Device made to move from end to end of the TV
   - Doppler shift changes sign \( \Rightarrow \) indicates the device position in front of speaker
1) Compute distance $D$ from these two positions of sign change.

2) Initial location of phone:
   - Use particle filter.
   - If particle must jump big to reach next position, filter away those particles.

3) Single Speaker System:
   - Use 1 speaker + WiFi or BLE
   - Apply AoA like techniques using phases $\phi_1$ and $\phi_2$ from sound and BLE.
   - Integer ambiguity doesn't arise because sampling fast (i.e., multiple times before $1\sigma$ movement).
   - CFO a major problem for RF phase tracking
   - Connects same clock between Sender & Receiver for precise phase.
3 Points of discussion:

1. Note that Doppler accumulates error since it gives velocity, which leads to location upon integration (FMCW doesn’t).

2. 44000 point FFT takes a second... not feasible for real time tracking. Thus STFT (short term Fourier transform) is used. Basically:
   1. Take fewer samples in real time
   2. Pad with 0’s to make 44000 samples
   3. Apply Hanning window for smoothing
   4. Take 44000 point FFT

   You get faster rate of Doppler estimates but preserve 1Hz freq. resolution
Questions?