End of Fall 2022 Semester Report

System Design team

1 Introduction

In this semester report the Systems Design team will detail some of what we learned over the fall 2022 semester. We will start with an overview of some relevant fundamentals of wireless channels in Section 2 which will be used to motivate following sections. We then explore the OFDM data transmission method in Section 3 and discuss some of the issues it is vulnerable to. With context on OFDM, we then provide an analysis of the main paper *Decimeter-Level Localization with a Single WiFi Access Point* [1] we read this semester in Section 4. We conclude by discussing potential uses for an RF environment simulation software called NYUSim in Section 5.

2 The Wireless Channel

Before diving into how the system works, it is important to understand what kind of data the system takes in to localize hosts. There are two types of data that are accessible in commercial WiFi chips that will be covered: RSSI and CSI.

2.1 RSSI

It is important to briefly mention that as a signal propagates through the air from the transmitter to the receiver, it can travel through multiple different paths. This idea is called multipath, and causes various issues in the system that will be mentioned later in this report. For now, it is only important to understand that any transmitted signal travels through multiple paths, picking up different phase offsets and weakening in power as the path length increases.

At any given point of time, the measured signal voltage at the receiver is:

$$V = \sum_{i=1}^{N} ||V_i|| e^{-j\theta_i}$$

Where V_i is the voltage and θ_i is the phase offset of the i^{th} path of N paths. RSSI data is this received power in decibels, so

$$RSSI = 10log_2(||V||^2)$$

Because RSSI values are essentially only the signal strength, localization methods utilize the idea that stronger signals imply the host to be closer to the access point. However, RSSI values are incredibly susceptible to change. The multiple paths that the signal propagates through significantly impact the total power of the system. When the paths that transmitted signals propagate through change, the RSSI value is heavily influenced as well. This naturally brings up one of the flaws of RSSI - the incapability to reflect the multiple paths that any transmitted signal travels through.

2.2 CSI

The wireless channel can be modeled as a filter, called the Channel Impulse Response (CIR).

$$h(\tau) = \sum_{i=1}^{N} a_i e^{-j\theta_i} \delta(\tau - \tau_i)$$

Where a_i is amplitude, θ_i is phase offset, and τ_i is time delay of the i^{th} of N paths. $\delta(\tau)$ is the Dirac delta function, so $\delta(\tau - \tau_i)$ is an impulse of width τ_i from time τ. Because of this, each component being added into the summation represents each path that the transmitted signal propagated through.

As mentioned in the previous subsection, the signal to be transmitted is sent through a carrier signal. The received signal ends up being the convolution between the transmitted signal and the CIR of the wireless channel. CIR of the wireless channel can be found by breaking down the received signal.

$$r(t) = s(t) \circledast h(t)$$

Where $r(t)$ is the received signal, $s(t)$ is the transmitted signal, and $h(t)$ is the wireless channel impulse response.

$$\mathcal{F}\{r(t)\} = \mathcal{F}\{s(t) \circledast h(t)\}$$
$$R(f) = S(f) \times H(f)$$

Because convolution is an arduous task for processing units, it is usually processed in the frequency domain, as convolution turns into multiplication. Thus, the Fourier Transform is taken for both sides. $R(f)$ is the Fourier Transform of the received signal, $S(f)$ is the Fourier Transform of the transmitted signal, and $H(f)$ is the Channel Frequency Response (CFR), or the Fourier Transform of the CIR.

$$H(f) = S^{-1}(f) \times R(f)$$
$$\mathcal{F}^{-1}\{H(f)\} = \mathcal{F}^{-1}\{S^{-1}(f)R(f)\}$$
$$h(t) = \mathcal{F}^{-1}\{S^{-1}(f)R(f)\}$$

Taking the inverse Fourier Transform after solving for the CFR, the CIR of the wireless channel can be obtained.

Modern day WiFi chips like the Intel 5300 card that will be highlighted throughout this paper give data values that are the sampled version of the CFR, called Channel State Information (CSI), available for use. Each sample of the CSI can be modeled as:

$$H(f_k) = ||H(f_k)||e^{\angle H(f_k)}||$$

where $H(f_k)$ is the CFR of the signal that was transmitted through sub carrier frequency f_k. It is clear that the phase offset incurred throughout the signal's lifetime can be easily extracted from CSI values. Specifically for localization, the time delay that is ingrained in this phase offset is useful to find time of flight of the signal:

$$\angle H(f_k) = -2\pi f \tau$$

2.3 Generating a Signal Using a Phase-Locked Loop (PLL) [2]

The clock signal which provides a signal at the center frequency of a transmitter or a receiver is often times generated by a PLL since a PLL will provide a clean output signal. The fundamental operating loop of a PLL looks somewhat like the following algorithm

Algorithm 1 Basic PLL Algorithm

```
1: out ← output signal
2: clk ← reference clock
3: while do
4:     raw_err ← clk ∘ out                    ▷ compute error between output and reference phase
5:     err_voltage ← ApplyControl(raw_err)    ▷ apply control logic to computed error
6:     out ← err_voltage ∘ out                ▷ feedback loop part
7: end while
```

Note that a PLL is an analog machine, not digital as this algorithm might suggest – this algorithm is simply for clarity purposes. When the same PLL is used to generate the signal for a variety of frequencies, the PLL must re-lock onto the frequencies as it changes. This takes a small amount of time due to the feedback loop nature of the PLL as can be seen in the algorithm above. Because of this, while the frequency of the signal coming out of the PLL is guaranteed to be a constant value, the phase of the signal coming out it not guaranteed to be the unvarying across the output signals as the phase is simply (normally) that of the reference signal. As we will see in Section 4 where we discuss Chronos [1], these dynamic phase differences across frequencies must be dealt with in a special way by the localization algorithm.

3 OFDM

3.1 Description of OFDM

Orthogonal frequency-division multiplexing (OFDM) is a type of digital modulation that is used in a wide range of communication systems, including wireless and wired networks. OFDM is a highly efficient and robust method of transmitting data over a wide frequency band. One of the key features of OFDM is that it divides the available frequency band into a large number of narrow sub-carriers, which are orthogonal (perpendicular) to each other. This allows multiple data streams to be transmitted simultaneously on different sub-carriers, increasing the overall data capacity of the system. OFDM also can adapt to changing channel conditions, which makes it resistant to interference and fading.

OFDM is widely used in a variety of communication systems, including wireless local area networks (WLANs), digital television (DTV) systems, and 4G and 5G mobile networks. For example, the IEEE 802.11a/g/n/ac WLAN standards all use OFDM as their modulation technique, and OFDM is also the primary modulation technique used in the European DVB-T and DVB-T2 DTV standards. In the field of mobile communications, OFDM is used in the 3GPP Long Term Evolution (LTE) and 5G NR standards for high-speed data transmission.

In these applications, OFDM provides several benefits over other modulation techniques. For example, in WLANs, OFDM allows for high data rates and improved range, which makes it well-suited for applications such as streaming video and online gaming. In DTV systems, OFDM allows for the efficient transmission of high-definition video and audio signals, which is essential for providing high-quality viewing experiences to users. In mobile networks, OFDM enables high-speed data transmission and improved

spectral efficiency, which is crucial for supporting the growing demand for data-intensive services such as mobile video and cloud computing.

Overall, OFDM is a key enabling technology for many modern communication systems, and its widespread use highlights the importance of this modulation technique in today's connected world.

3.2 Transmission model of OFDM signals

In terms of transmission, OFDM signals are typically generated using a discrete Fourier transform (DFT) to convert the data into the frequency domain. The data is then mapped onto the individual sub-carriers, which are combined to create the OFDM signal. The signal is then transmitted over the communication channel, where it can be recovered at the receiver using a reverse DFT process.
There are some parameters that may cause disturbances in the receiver. In the context of OFDM signal transmission, the time T' at which the receiver samples the signal cannot be assumed to be the same as the time T at which the transmitter sends the signal. Similarly, the carrier frequency oscillators used for modulation and demodulation may have a small frequency offset relative to the transmission bandwidth. This frequency difference can be modeled as a time-varying phase offset $\Theta_o(t)$ at the receiver. Because the receiver is unaware of the transmitter's time scale, the OFDM symbol window used to remove the guard interval will often be offset from its ideal setting by a time epsilon T delay. This delay can be incorporated into the channel model, resulting in the effective channel h_ϵ, i relevant to the receiver's time scale.

$$h_\epsilon(\tau, t) = h(\tau, t) * \delta(\tau - \epsilon T)$$

From the flow chart below, we could understand the transmission model more easily.

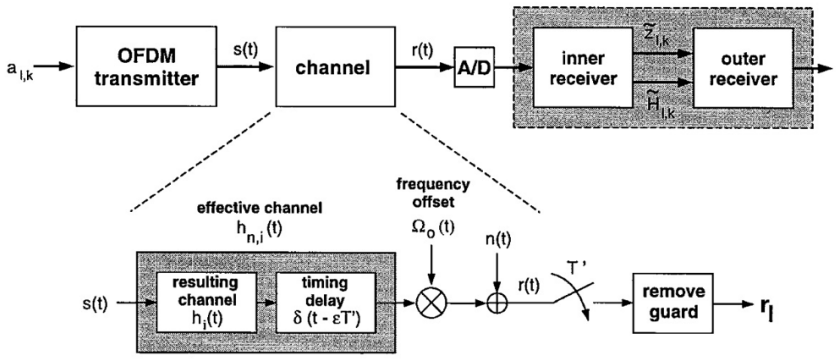

Figure 1: Complete baseband transmission model

With the information stated above, we could get the equation for the receiver input signal

$$r(t_n) = e^{j2\pi\Theta_o(nT')} \sum_i h_{\epsilon,i}(nT') * s(nT' - \tau_i) + n(nT')$$

and the lth received OFDM symbol could be represented by

$$r_{l,n} = r((n + N_g + l * N_s) * T')$$

[3]

3.3 How OFDM tackles undesirable conditions

Due to the fact that OFDM uses subcarriers, the wireless channels can be measured independently from each of them. With the characteristic of orthogonality between consecutive OFDM symbols, the time offset can be derived to be $\hat{\varepsilon} = N \hat{f}/2$. However, this is just a rough estimate when considering disturbance in various cases. For example, the allowable timing offset is assumed to be within a range, but there is no single ground truth synchronization point. With different size of maximum channel dispersion, there may be multiple solutions for the time index to preserve the orthogonality of the system. There are many existing algorithms for subcarrier frequency offset estimation, and some of them correlates with the estimation of the zero subcarrier offset[4]. The paper focused more on the detection of packet delay rather than time-of-flight because the former casts a major influence. But in the real case, if the measurement of time-of-flight was undesirable, OFDM provides a full tool set to help.

According to the Decimeter-Level Locolization paper, the concept of packet detection delay and time-of-flight delay was clearly separated in the estimations. The packet detection delay of zero-subcarrier can be neglected in calculation and the time-of-flight delay can be derived with results from all WiFi frequency band and the Chinese Remainder theorem.

All the above results were based on the measured channel of the zero-subcarrier, but it cannot be measured directly. Since the DC term falls on the zero-subcarrier and there is no data transmitted with it[5]. The OFDM modulation is especially sensitive to a DC offset due to self-mixing of the local oscillator in the radio circuit. This leads to a constant term that is superimposed on the useful signal components going into the ADC and subsequently into the DSP. In particular, if there is a DC offset as well as residual frequency offsets in an OFDM system, the unwanted signal components can overlap at least partially on the lower OFDM subcarriers, leading to significant performance degradations. Much effort has been put into the implementation of solving this problem with a better implementation such as the use of DC offset calibration or AC coupling, but few of them finally meets the desirable goal. In the paper, the team takes advantage of the characteristic that in the indoor scenario, OFDM subcarriers are continuous, which means the phase at zero-subcarrier can be recovered from the following subcarriers. With least square estimator to be the matrices, Low-pass interpolation performs better in channel frequency response estimation than other studied interpolation algorithms.

Other than what was mentioned in the Decimeter-Level Localization paper, OFDM provides solutions for many other problems, such as sampling clock frequency offsets and time-selective fading. Some of the undesirable cases may not be significant in the indoor case, but it is always beneficial to be careful of the potentially existing errors.

3.4 Advantages and disadvantages of OFDM

Advantages

- Traditionally, in a single-channel scheme, the data is sent sequentially, However, OFDM transmits data through multiple different channels, and the signal time on each subcarrier is longer than that on a single-channel system at the same rate which makes OFDM less susceptible to impulse noise.
- The OFDM adaptive modulation mechanism enables different subcarriers to use modulation methods independently based on channel conditions and noise backgrounds, enabling more efficient data transmission.

- OFDM has more resistance to symbol interference because of the implementation of the cyclic prefix.

Disadvantages

- OFDM is sensitive to frequency offset and phase noise. OFDM distinguishes different channels depending on the orthogonality between channels. Frequency offset and phase noise will ruin the orthogonality of channels.
- OFDM has a high peak to average power ratio (PAPR), making high demands on linearity in amplifiers.

Why OFDM

- Compared with FDMA, OFDM allows channels to space very close together with no overhead as that in FDMA.
- Compare with TDMA, there is no need for users to be time multiplex in OFDM. Therefore, there is no overhead associated with switching between users.

4 Localization with CSI

In accordance with the Decimeter-Level Localization paper, localization is possible with the CSI values retrieved from a single WiFi access point. This is enabled by a set of equations taking in the information from the CSI and translating it into time of flight. This is all done through the Chronos system, which is the researcher's solution to this problem. The reason for creating a system which can produce localization over WiFi technology is to allow this information to be used in everyday locations. Some of these beneficiaries are small businesses who want to ensure only their customers are receiving their free WiFi by Geo-Fencing the approved areas, smart homes which track where an occupant is in a house to provide more relevant assistance, and device to device localization, allowing two smart phones to track each other without having a WiFi signal. CSI is not a perfect tool and has many disadvantages such as hardware restrictions, calibration issues and phase offset. The Chronos system combats these issues and creates an accessible localization model through the use of CSI.

4.1 How is Localization Possible

The Chronos system takes in the CSI values from a WiFi enabled device and uses them to calculate time of flight, which in turn allows the system to localize a device from the signal. Because Chronos operates on an extremely wide band of channels, the channels must be stitched together in order to calculate time of flight. Chronos takes advantage of the fact that signals accumulate a phase depending on their frequency over time, so by using the equation below across all signals the Chinese Remainder theorem can be applied to calculate the LCM.

$$\forall i \in \{1, 2, \ldots, n\} \quad \tau = -\frac{\angle h_i}{2\pi f_i} \mod \frac{1}{f_i}$$

The LCM of this equation results in the time of flight that aligns all of the channels together, essentially finding the result shared across all of the channels which represents the true time of flight of the signal across the wide band. As mentioned earlier, hardware is a limitation when using CSI to calculate time of

flight. The main problem with the hardware is detection delay. This is the difference in time from the over air signal compared to when the receiver actually detects the packet. To combat hardware discrepancies, Chronos finds the true channel (subcarrier 0) which is claimed to have equivalent over the air and measured channel information. The measured channel can be calculated by the equation seen below which calculates the total measured channel phase at subcarrier k by taking the initial subcarrier phase and subtracting the phase rotation which is the detection delay that the hardware creates.

$$\begin{aligned}\angle\tilde{h}_{i,k} &= (\angle h_{i,k} + \Delta_{i,k}) \mod 2\pi \\ &= (-2\pi f_{i,k}\tau - 2\pi(f_{i,k} - f_{i,0})\delta_i) \mod 2\pi\end{aligned}$$

By taking the measured channel above and setting k to be equal to 0, the 0 subcarrier is equivalent to the over air measurement as well. This information is great, but the 0 subcarrier does not actually contain data. In order to calculate the data inside of the 0 subcarrier Chronos must interpolate the phase across all subcarriers to estimate the missing phase at the 0 subcarrier level. The result of this process is the packet detection delay being eliminated by utilizing the zero subcarrier. This process is termed as calibrating the system in order to get more accurate localization. This solution is far from perfect, and will continually vary depending on the hardware that is being used.

4.2 Combating Multipath

In order to solve the issue of time-of-flight spread, a result of RF signals bouncing off indoor interferences, Chronos obtains several copies of the signal, each with its own time of flight, so that it can compute the time-of-flight's direct path of the wireless signal apart from all other paths. These paths can be separated by using the inverse discrete Fourier Transform. After separating the direct path, the received signal is a result of multiple signals, each having a different propagation delay. This is seen in a multipath profile, which displays propagation delays so that it may be distinguished from other signal paths. Using the discrete Fourier transform, we can compute the inverse, which will yield the propagation delay for all paths and compute the multipath profile. Because the channel measurements are not uniformly spaced, we must use non-uniform discrete Fourier transform (NDFT). However, using this delivers many solutions, so Chronos adds the sparsity constraint, which favors solutions that are most dominant.

5 Experiments

OFDM channel-state data was required to test our implementation of the time-of-flight and localization algorithms described above. Acquiring empirical OFDM channel-state data is costly, as low-level access to software-defined radio information, as well as low-noise environments, are needed to collect accurate data. We instead turned to simulate channel-state for different scenarios. We attempted to use two tools for this task: a channel-state simulator developed by Sun et all at New York University, NYUSim, and Matlab's RF Toolbox combined with raytracing to simulate multipathing at the channel level.

5.1 NYUSim

NYUSim is a channel state simulator that allows users to simulate the channel characteristics of OFDM transmitters and receivers in changing scenarios. The user is able to configure multiple channel characteristics such as transmission frequency, bandwidth, and transmitter power, among others. It also allows the user to set spatial consistency constraints on the transmitter and receiver, being able to simulate the effects of the receiver moving in relation to the transmitter. Our goal was to use this tool to simulate channel-state

level information so we could recreate the time of flight determination and multipathing algorithms previously presented.

We first attempted to simulate a simple OFDM transmitter receiver case to familiarize ourselves with the tool and understand its outputs. We simulated the receiver being static at 10 meters from the transmitter.

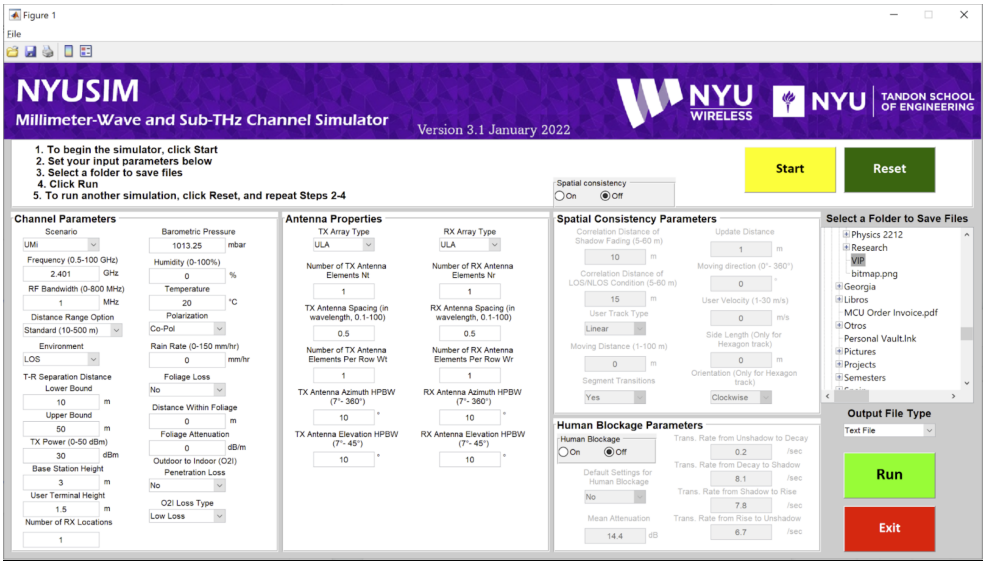

Figure 2: NYUSim simulation menu configured for initial case

The simulation outputs were underwhelming. We were able to extract power spectrums regarding the relative positions of the antennas on the transmitter and receiver as well as channel power spectra as a function of propagation time, but the results for a static receiver were the ones expected from a naive simulation instead of a channel-state based simulation. The generated power graph as a function of time contains a single spike at the naive time-of-flight solution for the situation, found by dividing the distance between transmitter and receiver by the speed of light.

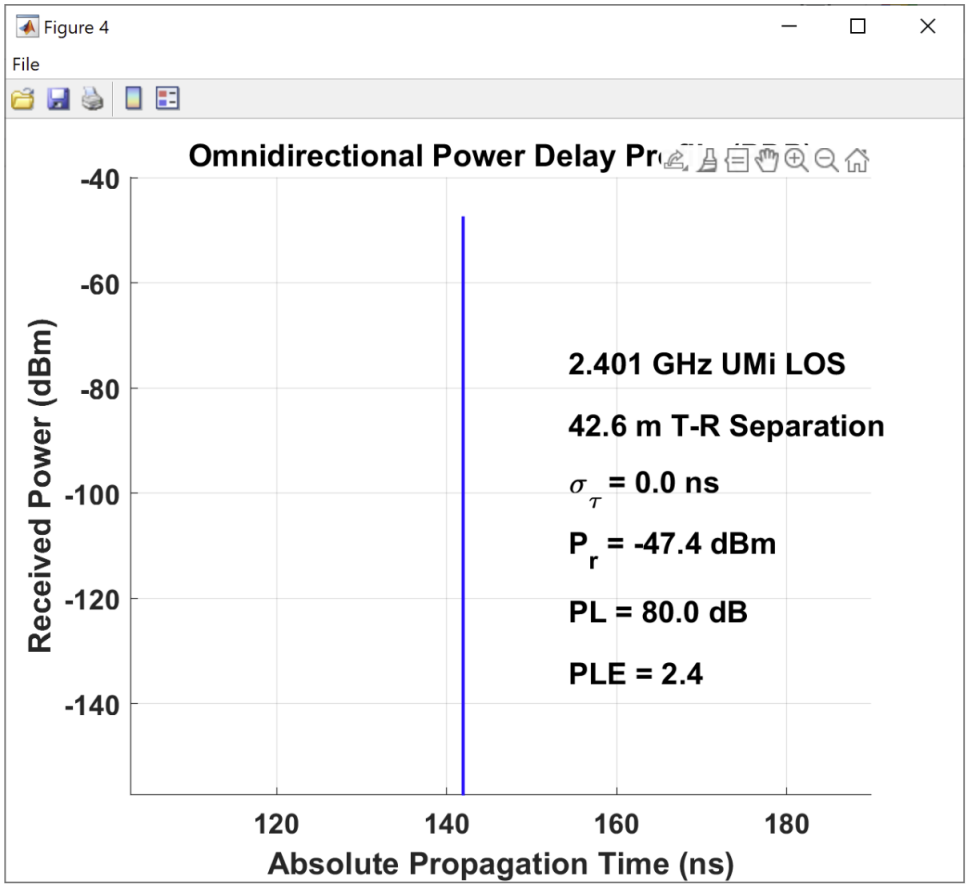

Figure 3: Estimated receiver power as a function of time-of-flight

From this simulation, we discovered NYUSim would not provide the channel-state information we were looking for to test our localization algorithms. We were able to extract time-of-flight information for simulated conditions, but we were not able to access channel-state information to apply our own algorithms to determine time-of-flight. We thus returned to search for a viable simulation solution.

5.2 MIMO-OFDM Channel-State simulation using Raytracing

We approached the problem of simulating channel-state information, this time using an approach developed using MatLabs Communications Toolbox and their ray tracing engine for wireless communication. This toolbox allows the user to define transmitter and receiver sites, much as with NYUSim, and to simulate the communication signals between both sites in a variety of physical scenarios using raytracing. We hope that this approach can emulate the multipathing present in real OFDM transmissions as well as give us access to lower-level information to test our time-of-flight determination algorithms on.

We started by using the provided example scenario to simulate a transmitter and receiver communicating inside a conference room. The simulation then computes the approximate multipathing response between the two devices using raytracing, and outputs a series of results.

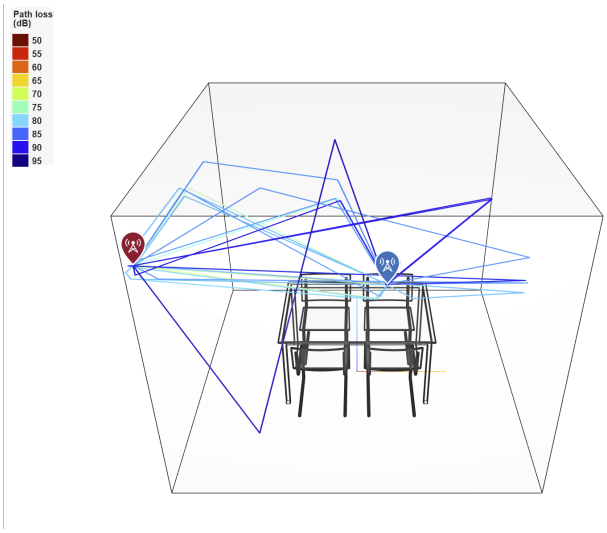

Figure 4: Simulation of OFDM communications inside a conference room using raytracing

The results from using this simulation approach seem promising. The simulator was able to produce non-naive power spectrums for the receiver as a function of signal delay, which after processing could serve as the input for our time-of-flight determination algorithms. Channel-state information is also provided by the simulator.

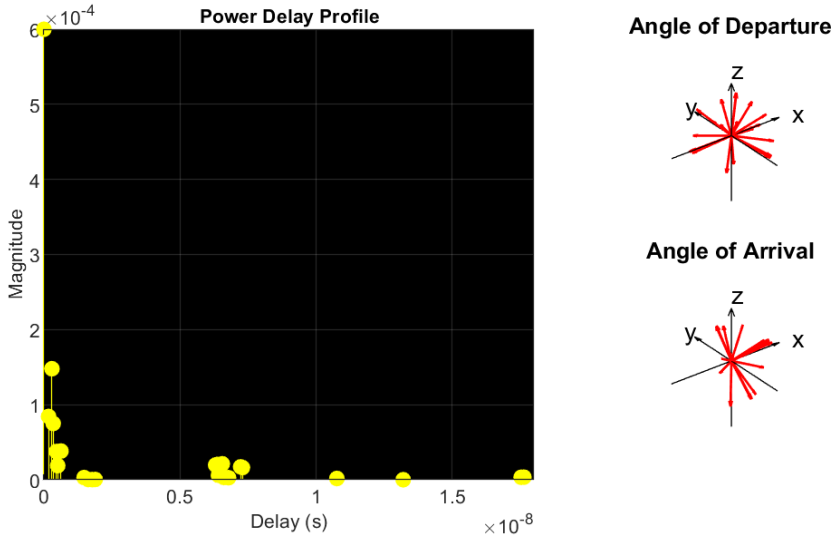

Figure 5: Transmitter Power Delay Profile and Signal Departure/Arrival Angles

For future progress on this simulator approach, more complex cases of transmitter-receiver pairs would need to be simulated in a variety of scenarios, and their outputs could then be used to demonstrate the validity of our time-of-flight estimation approach.

References

[1] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single wifi access point," in *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*, ser. NSDI'16. USA: USENIX Association, 2016, p. 165–178.

[2] E. Notes, "Pll phase locked loop tutorial amp; primer." [Online]. Available: https://www.electronics-notes.com/articles/radio/pll-phase-locked-loop/tutorial-primer-basics.php

[3] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broad-band systems using ofdm. i," *IEEE Transactions on Communications*, vol. 47, no. 11, p. 1668–1677, 1999.

[4] Z. Zhang, K. Long, M. Zhao, and Y. Liu, "Joint frame synchronization and frequency offset estimationin ofdm systems," *IEEE Transactions on Broadcasting*, vol. 51, no. 3, pp. 389–394, 2005.

[5] J. Heiskala and J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*. USA: Sams, 2001.

[6] D. Tse and P. Viswanath, "The wireless channel," in *Fundamentals of wireless communication*. Cambridge, 2013, p. 10–48.

[7] Z. Yang, K. Qian, C. Wu, and Y. Zhang, "Understanding of channel state information," *SpringerLink*, Oct. 2021.

[7]