Wi-Fi Bat–Echolocating a Device Connected to a Wi-Fi Transmitter Using Wi-Fi Signals

Alexander Horngacher, Jacob Larsson, David Lu, Noah Remmelgas and Emil Valfridsson

Bachelors thesis at The Department for Electrical Engineering, E2
Chalmers University
May 2025

Abstract

This thesis presents the design of a Wi-Fi sensing system that is capable of estimating the distance between two devices using Wi-Fi signals. The system leverages channel state information (CSI) extracted from a network interface controller (NIC) as well as the received signal strength indicator (RSSI). In this system, a Raspberry Pi equipped with a compatible NIC served as the receiver and a transmitter which transmitted Wi-Fi packets at varying distances. The thesis also discusses the limitations of such a system as signal noise and environmental multipath effects, and reflects on the ethical implications of Wi-Fi-based sensing with regard to privacy.

Contents

1 Abbreviations

Abbreviations	Definitions
CSI	Channel State Information
NIC	Network Interface Card/Controller
IoT	Internet of Things
OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
SNR	Signal to Noise Ratio
RSSI	Recieved Signal Strength Indicator
LoS	Line of Sight
ToF	Time of Flight
CIR	Channel Impulse Response
IFFT	Inverse Fast-Fourier Transform
MUSIC	MUltiple SIgnal Classification
FILA	Fine-grained Indoor Localization

2 Introduction

Over the past decades, wireless communication has dramatically evolved to become an essential part of modern society [1]. Recently interest has expanded into the field of "Wi-Fi sensing", a technology enabling the localization and detection of devices within a specific area by analysing signal strength and phase information from Wi-Fi networks [2].

The goal of this project is to locate a single device based on Wi-Fi signals received from another device by collecting and analysing relevant data. The distance will then be calculated by processing that data.

The scope of the project has been limited to the localization of a single device to avoid issues related to interference and noise, with a focus on achieving a high degree of accuracy within specified margins of error. This report outlines the background, methodology, results, and a discussion regarding future opportunities and potential improvements of the system.

2.1 Applications

As the amount of devices under the Internet of Things (IoT)[3] umbrella term has increased, more opportunities to apply Wi-Fi sensing has been created. There is, for example, a startup Company developing a Wi-Fi lightbulb that can track breathing and detect people falling, assisting elderly people to continue living at home instead of in retirement homes [4]. Although it has not been released yet, the potential for Wi-Fi sensing seems very promising. An already installed IoT technology in households can support Wi-Fi sensing without requiring extra hardware or sensors to be installed. Thus, Wi-Fi sensing is a promising candidate for applications like elderly assistance and indoor tracking, without replacing existing hardware.

Another application of Wi-Fi sensing has been explored to prevent wildlife accidents in rural areas [5]. Figure 1 below shows the setup to detect wild animals and pedestrians about to cross the road where a vehicle is driving.

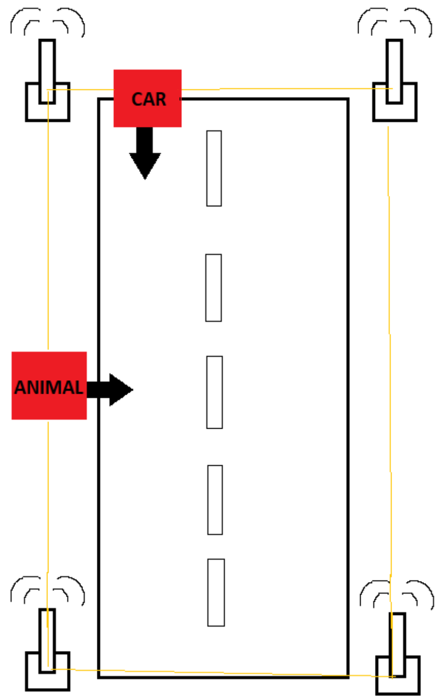

Figure 1: Wi-Fi sensing setup

From the survey [5], more than 1.35 million people die in road traffic accidents and is the main killer of people aged 5 to 29 years [5]. A substantial portion of these accidents are caused by collisions between drivers and wild animals and not only do people lose their lives, but these collisions cause a large cost for health care of injured. The project [5] show that Wi-Fi sensing can be a scalable and cost efficient implementation of preventing wild life collisions.

2.2 Background

Wireless sensing has gained a lot of attention in recent years due to potential applications in various areas such as indoor localization, security, healthcare, and smart home systems. Traditional localization methods such as GPS are usually ineffective indoors due to poor signal penetration and accuracy. However, now since a majority of households have a router and a Wi-Fi connection, methods utilizing Wi-Fi have become increasingly attractive [2].

Wi-Fi sensing utilizes signal properties like RSSI, CSI, etc. To detect changes in the environment. In particular CSI provides fine-grained information about the channel [6], which enables motion and presence detection even without requiring LoS.

2.2.1 How does Wi-Fi work?

The term Wi-Fi is an umbrella term to describe the various networking protocols contained in the IEEE 802.11 group of standards, however these protocols all work on similar principles. The information is transmitted on a carrier wave often referred to as the Wi-Fi band with the bitstream split into several streams sent over subcarriers, this is done via Orthogonal Frequency-Division Multiplexing (OFDM) in the case of 802.11 a, g, n, ac, ah, HIPERLAN/2 while Wi-Fi 6 (802.11 ax) uses Orthogonal Freuency Division Multiple Access (OFDMA) instead [7].

2.2.2 CSI

CSI is information in the form of a complex number that shows how the signal propagates from the transmitter to the receiver. During the CSI extraction from an NIC, not only the CSI can be retrieved, but also RSSI and several other parameters which helps with localization.

2.2.3 RSSI

RSSI provides valuable information regarding the received signal power. It is measured in dBm (decibel-milliwatts)[8]. Wi-Fi signals are susceptible to physical obstacles, primarily in the form of large scale fading such as path loss, shadowing and from small scale fading effects such as multipathing [9]. By analysing the RSSI, a model for distance estimation can be created through path loss.

2.2.4 Wi-Fi Sensing

Wi-Fi sensing relies on analysing the CSI extracted from Wi-Fi signals. Considering that the wireless channel is influenced by physical obstacles and environmental changes, the CSI-data can detect movement in a room, like elderly people falling, and track breathing as stated in the background however, this project will be limited to distance estimation [5]. In the next sections, different methods for distance estimation will be described and their corresponding advantages and limitations.

2.2.5 Distance Estimation

Distance estimation in Wi-Fi sensing relies on analysing the characteristics of the wireless signal as it travels between the transmitter (a laptop in this project) and receiver (Raspberry Pi). One of the key signal features used is the CSI, which provides detailed information about how the signal propagates across multiple subcarriers in the Wi-Fi channel.

A precise approach is to analyse the phase and amplitude data within CSI. The phase shift of the signal can be used to estimate the Time of Flight (ToF), i.e. the time it takes for the signal to travel from the sender to the receiver.

One method to estimate the ToF is to analyse the CSI data in the time-plane, following the method described in Hands-on Wireless Sensing with Wi-Fi: A Tutorial [10]. In order to estimate the ToF this method relies on the channel impulse response (CIR) which can be accessed by taking an inverse fourier transform on the CSI. Under the assumption that the peak response corresponds to the shortest path and that the shortest path is a clear LoS the subcarrier with the largest response can then be identified by analysing the CIR, the ToF for the subcarrier is then estimated as: the index of the subcarrier divided by the bandwidth of the wave [10]. This method however has its drawbacks, the resolution of the estimation is limited due to bandwidth limitations and if the collected data is noisy or was captured without LoS the estimation will be far from accurate, considering the propagation velocity of the Wi-Fi signals [10].

Another method to estimate distance is to use the free space path loss that wireless signals experience as they propagate through the air which can be described by Friis transmission equation.

$$\frac{P_t}{P_r} = D_t D_r \left(\frac{\lambda}{4\pi R} \right)^2 \tag{1}$$

where P is the power of the signal and D is the gain and efficiency factor of the antenna with subscripts t and r representing the receiving and transmitting antenna [11].

2.3 Hardware and Software

2.3.1 Network Interface Cards

The NIC is the hardware that allows a computer to connect to a network; they do it by providing the device with a dedicated point of connectivity to a network. NICs then transmits and receives data packets between the network and this dedicated point in the device. They convert data from analog to digital when reveiwing packets from the network and converts data from digital to analog when transmitting to the network. They also manage data flows to control traffic and maintain optimal network performance [12].

In order to be able to extract the CSI data from the NIC, it must allow for CSI extraction. Examples of such cards are: BCM43684, Intel AX200, Intel AX210 and Atheros 802.11n. For this project, the Intel AX210 was selected.

2.3.2 FeitCSI

FeitCSI is an open source CSI-extraction tool that is compatible with the AX210 and AX200 NICs developed by Kuskosoft and is the main software used in this project [13].

FeitCSI has 2 functions: measure and inject. Inject makes the NIC actively transmit Wi-Fi packets that can be configured with different settings such as modulation (MCS index), transmission power, guard interval, and number of spatial streams. Measure then passively captures the transmitted packets and extracts the CSI. The CSI is then saved to a text file for further processing.

In order to capture information and extract the CSI-data with FeitCSI, another computer must be transmitting data packets for FeitCSI to detect and measure.

2.3.3 Ubuntu

Ubuntu is a widely used open source operating system based on the Linux kernel. It is developed and maintained by Canonical Ltd. It is developed with user-friendliness and stability in mind [14].

In this project, Ubuntu was used on both the transmitter and the receiver. For the laptop 22.04.05 LTS (Long Time Support) was used and for the receiver, 24.04.02 LTS was used.

3 Purpose

The primary purpose for this project is to apply and test different methods of Wi-Fi sensing. There exist a wide range of various methods to calculate both distance and AoA. These methods are to be tested, documented and discussed later in this report.

The secondary purpose of this project is to create an effective Wi-Fi sensing system capable of accurately determining the location of a device based on the Wi-Fi packets it receives from another device. This system is also intended to address the limitation of traditional indoor localization methods by providing a reliable alternative using existing Wi-Fi infrastructure.

By achieving these target goals, the project demonstrates the Wi-Fi sensing technology for indoor localization tasks. Furthermore, the evaluation of the different methods highlights which approach is most effective.

4 Method

The initial goal of this project was to create a Wi-Fi sensing system, which is capable of accurately locating devices connected to a router through distance and angle estimation.
The project is divided into several parts:

1 Setting up the work environment with Raspberry PI, FeitCSI and the AX210 chip.

2 Setting up a laptop with Linux and FeitCSI.

3 Coding and processing the CSI-data extracted with FeitCSI and calculate the distance between the laptop and the Raspberry PI

4 Transmitting from different distances in order to calibrate and adjust the codes and calculations

4.1 Setup

When calculating the distance between the 2 devices, there was one Wi-Fi-transmitting device: a laptop that was injecting data frames with FeitCSI and the Raspberry Pi, which was receiving and measuring. The CSI data was then extracted with the help of a NIC that allowed for CSI extraction and processed in the Raspberry Pi. The distance was then calculated with a python script which would then output the distance and angle.

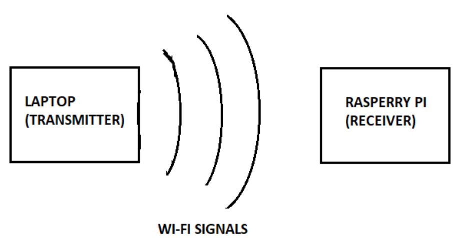

Figure 2: Block diagram of Wi-Fi sensing setup

Figure 2 shows a block diagram for the setup when measuring. The distance between the transmitter and receiver is changed for each measuring to get data from different set distances.

4.1.1 Hardware

The required hardware for this project, as previously mentioned was:

- Laptop with Ubuntu and FeitCSI setup

- Raspberry Pi with the AX210 NIC card mounted onto it with a PCie to M.2 HAT+ adapter
- Display, mouse and keyboard for proper operation of the Raspberry Pi

4.2 Extracting and Processing CSI Data from FeitCSI

After setting up the environment and configuring the NIC, FeitCSI was used to capture CSI data packets transmitted from the laptop to the Raspberry Pi. The captured CSI contains information for each subcarrier across the Wi-Fi signal's bandwidth, which can be leveraged for accurate localization.

The signal that is being broadcasted gets weaker and more distorted as it travels through space, particularly when encountering obstacles such as walls, furniture or human bodies. These interactions result in attenuation, phase shifts, and multipath propagation, all of which are captured in the CSI measured at the receiver.

The first measuring was to capture the CSI data. These measurements were taken in a big, open and empty 30x30 meter room. The receiver was placed in one of the corners and the transmitter was placed at distances ranging from 2.5 to 20 meters. In order to be able to accurately process the CSI and calculate the distance, many measurements are taken in order to average out errors, preferably as many samples as possible. In this project, about 300 packets were captured for each distance. The settings for the injection was:

Frequency - 5180 GHz

Bandwidth - 80 MHz

Format - HESU

The rest of the settings was left in the default FeitCSI configuration.

The measurements for the RSSI model were taken separately from the CSI measurements. These were done outdoors to minimize the effects of multipathing. The measurements were taken with the transmitter placed stationary 50 cm from the ground and the receiver being placed 50 cm from the ground at distances 1, 5, 10, 15, 20 m from the transmitter. Wireshark was then used on the receiver to capture the RSSI of the packets injected with FeitCSI on the transmitter.

4.3 Distance Estimation Method 1: RSSI

Distance estimation based on the RSSI converts the power loss that a packet experiences during propagation into a range estimate. The technique relies on empirical/experimental path-loss models that relate the distance to the average attenuation of the signal [15].

There are many different RSSI distance estimation models that can be used based on the conditions and accuracy needed, the method used for this project is a simple Path loss model.
The model can be described by equation (2)

$$P_r = P_t \left(\frac{d_0}{d} \right)^{\alpha} \tag{2}$$

where P_t is a reference power at a reference distance d_0 and P_r is the power received at distance d and α is the path loss exponent [16]. Equation (2) can then be rewritten as

$$10\log_{10}(P_r) = 10\log_{10}(P_t) + 10\alpha\log_{10}(d_0) - 10\alpha\log_{10}(d) \tag{3}$$

where $10\log(P_r)$ and $10\log(P_t)$ represent the RSSI in dBm at d and d_0, respectively. Using this $10\log(P_r)$ can represented with RSSI and $10\log(P_t)$ with Reference which gives the equations (4) - (6)

$$RSSI = Reference + 10\alpha\log_{10}(d_0) - 10\alpha\log_{10}(d) \tag{4}$$

$$\log_{10}(d) = \frac{Reference - RSSI + 10\alpha\log_{10}(d_0)}{10\alpha} \tag{5}$$

$$d = 10^{\left(\frac{Reference - RSSI + 10\alpha\log_{10}(d_0)}{10\alpha}\right)} \tag{6}$$

4.4 Distance Estimation Method 2: Fine-grained Indoor Localization

Fine-grained Indoor Localization (FILA) employs a free space path loss model to estimate distances using CSI [6]. After collecting CSI data, an inverse Fourier transform is applied to derive the Channel Impulse Response (CIR) of the signal. The CIR is then analysed to identify either the line-of-sight (LoS) component or the nearest non-line-of-sight (NLoS) path [6]. Based on this information, all signal components with amplitudes lower than 50% of the peak amplitude of the LoS or NLoS path are filtered out. This step is performed to reduce the influence of multipath reflections, which can introduce significant error in distance estimation [6].

Once the filtering process is complete, the CSI is reconstructed using a Fourier transform. The effective CSI power is then calculated using Equation (7):

$$\mathrm{CSI}_{eff} = \frac{1}{K}\sum_{k=1}^{K} \frac{f_k}{f_c} \times |A|_k \tag{7}$$

In this equation, K denotes the number of subcarriers, f_c represents the central frequency of the Wi-Fi band, f_k is the frequency at subcarrier k and $|A|_k$ is the amplitude of the CSI at subcarrier k [6].

Using a revised form of Friis' transmission equation, the distance d to the monitored device is then estimated by the following expression:

$$d = \frac{1}{4\pi}\left[\left(\frac{c}{f_c \times |\mathrm{CSI}_{eff}|}\right)^2 \times \sigma\right]^{\frac{1}{n}} \tag{8}$$

Here, c denotes the speed of light, σ represents the combined gain and efficiency factor of the antennas, and n is the environment-dependent path loss exponent. Both σ and n must be calibrated based on the specific hardware configuration and the physical environment in which the system is deployed [6].

4.5 Distance Estimation Method 3: CIR-based Time of Flight

The CIR-based distance estimation relies on estimating the ToF of the signal through the inverse fast-Fourier transform (IFFT) of the CSI data to convert it to the corresponding CIR [10]. The CIR for sub-carrier k can be described as:

$$h_k(t) = \alpha_k \delta(t - t_k) \tag{9}$$

where α_k is the amplitude and t_k is the time delay of the sub-carrier, this means that the CIR is directly correlated to the ToF of the sub-carrier[10]. The CIR samples correspond to discrete time delays with a resolution of $\frac{1}{bw}$ where bw is the bandwidth of the carrier wave, the ToF for sub-carrier k can then be estimated as: $\frac{k}{bw}$ [10]. By identifying the sub-carrier with the strongest path and assuming that the strongest path corresponds to a LoS transmission the distance between the transmitter and receiver can be estimated as the time-of-flight of this path multiplied by the speed of light c [10].

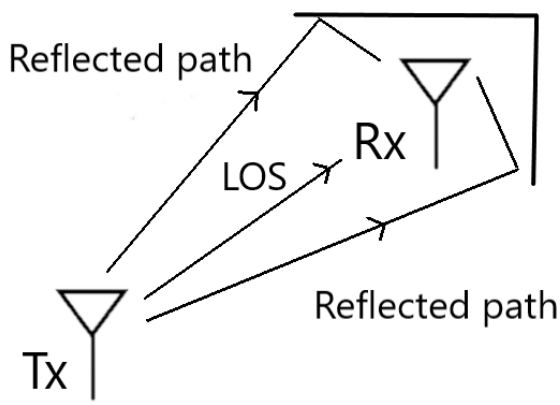

Figure 3: Illustration of multipathing showing a LoS transmission and two multi-paths.

Figure 3 presents a simplified illustration of three signal propagation paths from the transmitter to the receiver. Among these, two are reflected paths and one represents the line-of-sight (LoS) path. The reflected paths are expected to exhibit lower signal strength due to the longer distances they travel. Consequently, the LoS path can typically be identified by its more prominent peak in the frequency domain.

This method offers the benefit of being straightforward to implement; however, it comes with significant limitations in terms of reliability. It is based on the assumption that the strongest signal path corresponds to the LoS component, an assumption that does not always hold true [10]. In environments with extensive multipath propagation, the direct path may be attenuated due to obstructions or interference, while a reflected path may exhibit a stronger signal, thereby leading to incorrect identification of the true LoS component [10].

A further limitation of this approach lies in its low time resolution, which is inherently constrained by the available bandwidth. For example, the AX210 network interface card supports a maximum bandwidth of 160 MHz, which translates to a minimum resolvable distance of $\frac{c}{160\times10^6} = 1.875$ meters. This coarse resolution makes the method unsuitable for applications that require high precision in distance estimation.

4.6 Angle of Arrival Estimation

When trying to determine the location of a device, both distance and angle from the transmitter to the receiver is necessary. The image below is a simplified sketch of what the problem looks like.

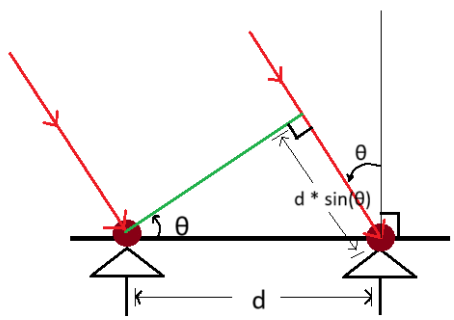

Figure 4: Angle estimation visual

The distance d represents the spacing between the antennas mounted on the transmitter/receiver. Here, θ represents the angle from the transmitter to the receiver. A key factor in solving the AoA is through the phase shift that the different antennas hear. Denoting the phase difference as $\Delta\phi$.

$$\Delta\phi = 2 \cdot \pi \frac{\Delta L}{\lambda} \tag{10}$$

Here ΔL denotes the difference in distance that the incoming signal has to travel to the furthest antenna, which in the figure is marked as $d \cdot sin(\theta)$. Multiplying the amount of extra wave lengths travelled $\frac{\Delta L}{\lambda}$, by an entire phase of $2 \cdot \pi$ radians, the phase difference between the antennas is acquire. By substituting $\Delta L = d * sin(\theta)$ one gets:

$$\Delta\phi = 2 \cdot \pi \cdot \frac{d \cdot \sin(\theta)}{\lambda} \tag{11}$$

The sine term is isolate and the resulting formula for estimating the AoA is acquire:

$$sin(\theta) = \frac{\lambda \cdot \Delta\phi}{2 \cdot \pi \cdot d} \tag{12}$$

$$\theta = arcsin(\frac{\lambda \cdot \Delta\phi}{2 \cdot \pi \cdot d}) \tag{13}$$

This situation is under the assumption that incoming signals are travelling the shortest path, which is called the LoS [17]. Transmitted signals propagate spherically into space and reflect on surfaces. In indoor localization, a lot of reflection surfaces exist. Therefore, several copies of the same signal arrive at the receiver from different angles and different distances travelled, this is called a multipath problem [18]. With a simple implementation that does not handle this issue, the spatial resolution becomes very low. This means that the copies of the same signal start to blend together and it becomes hard to distinguish the signals from one another and which signal has travelled in LoS, i.e the signal of interest. Multiple methods to solve this issue exist such as non-linear least squares approximation, which is used in other projects [10]. Another, more advanced, method is called MUltiple SIgnal Classification (MUSIC) [19].

5 Results

Measurements were taken with the transmit power of 1 dBm. RSSI measurements had 2457 MHz frequency, 40 Mhz Bandwidth with the format HT and beamforming turned off. FILA and ToF used 5180 MHz, 80 MHz on a HESU format with beamforming turned off though multiple other settings were tried with similar results.

5.1 RSSI Path Loss

The RSSI data collected yielded the following results:

True distance (m)	1	5	10	15	20
Measured RSSI (dBm)	-50	-61	-65	-68	-64
Calculated distance (m)	1	5.4	10.0	15.8	8.6

Distances were calculated using the formula from Equation (6) with d_0 set to 1 meter and α set to 1.5. α was obtained through trial and error. While the estimated distances for 5, 10 and 15 m are quite consistent, the results for 20 meters are not good. The probable reason for the measured RSSI being so high for this measurement in particular is due to the environment in which the measurements were taken. Due to the limited space available the 20 m measurement had to be taken with the receiver about 1.5 m from a wall introducing multipath disturbances. This was expected as RSSI measurements are very susceptible to environmental changes and reflective objects.

5.2 FILA

Using the FILA method in 4.4 on CSI data collected yielded the results in the table below.

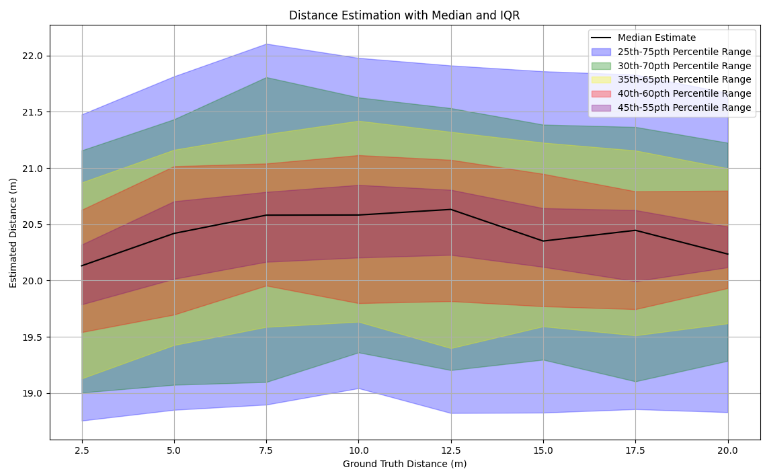

Figure 5: Graph showing the quantized distance estimated for each packet at each true distance

Actual Distance (m)	2.5	5.0	7.5	10.0	12.5	15.0	17.5	20.0
Calculated Distance	22.3	21.5	21.7	18.5	20.7	18.7	19.4	18.9

This method requires calibration of variables n and σ. But as there is no correlation between the actual distance and calculated distance it means calibration can't be meaningfully done and as such this method doesn't work with the CSI data that was collected.

5.3 ToF

Using the ToF code (see appendix) based on the ToF method described in section 4.4 on the CSI data collected yielded the following results:

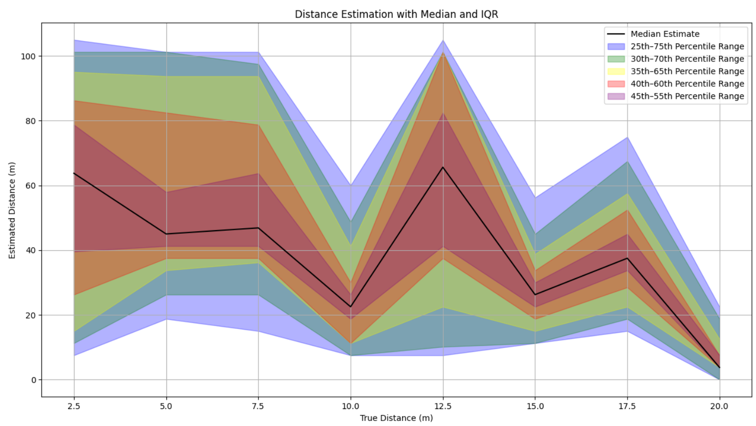

Figure 6: Graph showing the quantized distance estimated for each packet at each true distance

True distance (m)	2.5	5	7.5	10	12.5	15	17.5	20
Calculated distance (m)	63.8	45	46.9	22.5	65.6	26.3	37.5	3.8

It is likely that the data collected was too noisy and influence by multipath signals for this algorithm to accurately detect the correct peaks in the CIR, while there are measures that could have been implemented in order to clean up the data such de-noising/smoothing, because this data also yielded no usable estimation in the FILA method. It is likely the CSI data itself is the problem. Due to hardware issues no further CSI data could be collected.

6 Discussion

6.1 Problems and Limitations

During the project, there have been many challenges along the way. All of them have been limiting and delaying the scope of the project. A list of them are:

- Shipping delays
- Non-compatible PCIe to M.2 HAT+ adapter
- Problems with setting up the Raspberry Pi

- Raspberry Pi and FeitCSI breaking

6.2 Privacy Concerns and Ethical Challenges

Keeping track of the movement of citizens is considered a violation of privacy. With this in mind, the Wi-Fi Bat project could spark controversy considering the aim of the project is to locate devices connected to Wi-Fi, which the majority of the population have in their pocket at all times. Even if the Wi-Fi Bat project required the device being monitored to use an application to send information, this is a step in the direction of monitoring individuals through a widely used and almost essential technology in this age.

One way of justifying Wi-Fi sensing from the privacy concern is to separate sensing with identification. Wi-Fi sensing could map the movement of people in a room without identifying who is being monitored. This could be compared to motion detectors that turn on lights or open doors. That is rather tracking of movement, not tracking of individuals.

However, by disregarding the privacy concern, one provides a new technology to the world that can be used as a tool for applied use cases, which can lead to further innovation and use cases. This does not mean that the privacy concerns should be disregarding, rather the pros and cons of the technology needs to be carefully assessed.

6.3 Project Outcomes

This project successfully demonstrated that it is possible to estimate the distance between two Wi-Fi enabled devices using only RSSI. The objective was to explore practical applications of Wi-Fi sensing for indoor localization and develop a system capable of estimating distances between two devices, while a complete system was not developed the results of the RSSI path loss model shows that it is possible.

An RSSI-based distance estimation model was implemented, translating signal attenuation into a distance measurement. RSSI data was collected at a known distance to validate the performance of the model and evaluate its accuracy in real-world conditions. The results were promising under line-of-sight conditions, however further measurements would have to be taken with multiple distances in order to develop a more reliable model.

Throughout the project, extensive work was carried out to set up the technical environment, collect and process data, and validate the model. In addition to the technical implementation, the project provided insight into the limitations of RSSI-based methods, particularly their sensitivity to multipath effects and changes in the surrounding environment. These findings are important for informing future work in the field of Wi-Fi sensing and localization.

7 References

References

[1] T. Rappaport, "The wireless revolution," *IEEE Communications Magazine*, vol. 29, no. 11, pp. 52–71, 1991. DOI: 10.1109/35.109666.

[2] Z. Yang, Z. Zhou, and Y. Liu, "From rssi to csi: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, Dec. 2013, ISSN: 0360-0300. DOI: 10.1145/2543581.2543592. [Online]. Available: https://doi.org/10.1145/2543581.2543592.

[3] IBM, *Internet of things*, Accessed: 2025-04-22, n.d. [Online]. Available: https://www.ibm.com/think/topics/internet-of-things.

[4] M. T. Review, *How wi-fi sensing became usable tech*, Accessed: 2025-04-22, 2024. [Online]. Available: https://www.technologyreview.com/2024/02/27/1088154/wifi-sensing-tracking-movements/.

[5] S. V. Ducca, A. Jordão, and C. B. Margi, "Detection and classification of animal crossings on roads using iot-based wifi sensing," in *2023 IEEE Latin-American Conference on Communications (LATINCOM)*, 2023, pp. 1–6. DOI: 10.1109/LATINCOM59467.2023.10361871.

[6] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, "Fila: Fine-grained indoor localization," in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 2210–2218. DOI: 10.1109/INFCOM.2012.6195606.

[7] Keenetic Support. "What you need to know about Wi-Fi 6 (IEEE 802.11ax)." [Online; accessed 6-May-2025]. (2021), [Online]. Available: https://support.keenetic.com/eu/titan/kn-1811/en/20208-what-you-need-to-know-about-wi-fi-6--ieee-802-11ax-.html.

[8] Wikipedia contributors. "Received signal strength indicator." Accessed: 2025-05-13. (2024), [Online]. Available: https://en.wikipedia.org/wiki/Received_signal_strength_indicator.

[9] ScienceDirect Topics. "Shadowing effect." Accessed: 2025-05-13. (n.d.), [Online]. Available: https://www.sciencedirect.com/topics/engineering/shadowing-effect.

[10] Z. Yang, Y. Zhang, G. Chi, and G. Zhang, *Hands-on wireless sensing with wi-fi: A tutorial*, Jun. 2022. DOI: 10.48550/arXiv.2206.09532.

[11] J. A. Shaw, "Radiometry and the friis transmission equation," *American Journal of Physics*, vol. 81, no. 1, pp. 33–37, Jan. 2013, ISSN: 0002-9505. DOI: 10.1119/1.4755780. eprint: https://pubs.aip.org/aapt/ajp/article-pdf/81/1/33/12913789/33_1_online.pdf. [Online]. Available: https://doi.org/10.1119/1.4755780.

[12] TechTarget Contributor. “What is a network interface card (nic)?” Accessed: 2025-04-23. (2023), [Online]. Available: https://www.techtarget.com/searchnetworking/definition/network-interface-card.

[13] M. Hutar, P. Brida, and J. Machaj. “Feitcsi, the 802.11 csi tool.” (2023), [Online]. Available: https://feitcsi.kuskosoft.com.

[14] Canonical Ltd. “Ubuntu.” Accessed: 2025-04-22. (2025), [Online]. Available: https://ubuntu.com.

[15] T. Türkoral, Ö. Tamer, S. Yetiş, E. İnanç, and L. Çetin, “Short range indoor distance estimation by using RSSI metric,” *IU–JEEE*, vol. 17, no. 2, pp. 3295–3302, 2017.

[16] F. Y. M. Adiyatma, D. J. Suroso, and P. Cherntanomwong, “Regression-based path loss model correction to construct fingerprint database for indoor localization,” in *Proceedings of the 2023 6th International Conference on Electronics, Communications and Control Engineering*, ser. ICECC ’23, Fukuoka, Japan: Association for Computing Machinery, 2023, pp. 181–186, ISBN: 9798400700002. DOI: 10.1145/3592307.3592335. [Online]. Available: https://doi.org/10.1145/3592307.3592335.

[17] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, “Spotfi: Decimeter level localization using wifi,” *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 269–282, Aug. 2015, ISSN: 0146-4833. DOI: 10.1145/2829988.2787487. [Online]. Available: https://doi.org/10.1145/2829988.2787487.

[18] J. Xiong and K. Jamieson, “Arraytrack: A fine-grained indoor location system,” in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, ser. nsdi’13, Lombard, IL: USENIX Association, 2013, pp. 71–84.

[19] J. P. Merkofer, G. Revach, N. Shlezinger, T. Routtenberg, and R. J. G. van Sloun, *Da-music: Data-driven doa estimation via deep augmented music algorithm*, 2023. arXiv: 2109.10581 [eess.SP]. [Online]. Available: https://arxiv.org/abs/2109.10581.

8 Appendix

8.1 Code

```
def tof(csi_data,bw):
    """
    tof: Rough distance estimation using naive peak picking from
    ↪  IFFT of CSI.

    Parameters:
        csi_data (np.ndarray): CSI data of shape [packet,
        ↪  subcarrier, antenna]
```

```
        bw (float): Bandwidth in Hz

    Returns:
        dist_mat (np.ndarray): Estimated distance [packet,
        ↪  antenna]
    """
    c = 3e8  # Speed of light
    packet_num, subcarrier_num, antenna_num = csi_data.shape

    # Compute CIR using IFFT over subcarriers,
    # output shape [packet, ifft_point, antenna]
    cir_sequence = np.fft.ifft(csi_data, n=subcarrier_num,
    ↪  axis=1)

    # Average over the mirrored second half is not needed due to
    # cyclic repetition, just use the first half
    half_point = subcarrier_num // 2
    half_sequence = cir_sequence[:, :half_point, :]

    # Find the peak index for each packet and antenna
    peak_indices = np.argmax(np.abs(half_sequence), axis=1)  #
    ↪  [packet, antenna]

    # Calculate ToF
    tof_mat = peak_indices / bw  # [packet, antenna]
    # Calculate distance
    dist_mat = tof_mat*c
    return dist_mat

import numpy as np
import subprocess
import csi_read
import time
import FILA


def run_feitcsi():
    with open('/home/wifi/csi.txt', 'w') as f:
            f.truncate(0)
    command = "sudo feitcsi --mode measure --frequency 5180
    ↪  --channel-width 40 --format HT --output-file
    ↪  /home/wifi/csi.txt"
    process = subprocess.run(["gnome-terminal", "--", "bash",
    ↪  "-c", f"{command}; bash"], check=True)
    time.sleep(2)
    try:
```

```
        while True:
            for i in range(1):
                time.sleep(1)

                csi_matrix =
                ↪  csi_read.CSIRead.main('/home/wifi/csi.txt')

                FILA.distance.main(csi_matrix)
            with open('/home/wifi/csi.txt', 'w') as f:
                f.truncate(0)


    except ValueError as v:
        while True:
            print(v)
            time.sleep(2)

    except KeyboardInterrupt:
        process.terminate()
        return

run_feitcsi()


import numpy as np

class distance():
    #Signal variables
    freq = 5180e6
    subcarrier_num = 57
    subcarrier_freq = np.linspace(5160e6, 5200e6, subcarrier_num)
    c = 299792458
    bandwidth = 40e6

    def csi_to_cir(csi):
        return np.fft.ifft(csi)

    def cir_to_csi(csi):
        return np.fft.fft(csi)

    def FILA_distance(csi):
        cir = distance.csi_to_cir(csi)
        cir = cir[:,:,:1]
        cir_shape =
        ↪  np.array([len(cir[:,0,0]),len(cir[0,:,0]),len(cir[0,0,:])])
```

```python
    cirArgMax = np.argmax(np.abs(cir), axis=1, keepdims=True)
    cirMax = np.empty((cir_shape[0], 1,
        cir_shape[2]),dtype=complex)
    cir_filtered = np.empty((cir_shape[0], cir_shape[1],
        cir_shape[2]), dtype=complex)


    for i in range(cir_shape[0]):
        for j in range(cir_shape[2]):
            cirMax[i,0,j] = cir[i,cirArgMax[i,0,j],j]

    for i in range(cir_shape[0]):
        for j in range(cir_shape[1]):
            for k in range(cir_shape[2]):
                if np.abs(cir[i,j,k]) >=
                    0.5*np.abs(cirMax[i,0,k]):
                    cir_filtered[i,j,k] = cir[i,j,k]
                else:
                    cir_filtered[i,j,k] = 0

    csi = distance.cir_to_csi(cir_filtered)
    CSI_eff = np.mean(distance.subcarrier_freq /
        distance.freq * np.abs(cir))

    #omega and n variables tuned to the environment and
        equipment used
    omega = 8000000
    n = 2

    d_LOS = 1/(4*np.pi) * ((distance.c / (distance.bandwidth
        * np.abs(CSI_eff)))**2 * omega)**(1/n)
    return d_LOS

def main(csi):
    print(distance.FILA_distance(csi))
```