JAMES COOK UNIVERSITY

THESIS

Involuntary Indoor Localisation using WiFi

Author:
Jack NEWMAN

Supervisor:
Bronson PHILIPPA

College of Science and Engineering
October 6, 2016

Statement of Access

I, the undersigned, author of this work, understand that James Cook University may make this thesis available for use within the University Library and, via the Australian Digital Theses network, for use elsewhere. I understand that, as an unpublished work, a thesis has significant protection under the Copyright Act and I do not wish to place any further restriction on access to this work

Signed:

Date:

Declaration of Authorship

I, Jack NEWMAN, declare that this thesis titled, "Involuntary Indoor Localisation using WiFi" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____

Date: ____

Acknowledgements

A big thank you to my thesis supervisor Bronson Philippa, none of this would have been possible without him.

JAMES COOK UNIVERSITY

Abstract

College of Science and Engineering

Computer System Engineering

Involuntary Indoor Localisation using WiFi

by Jack NEWMAN

WiFi localisation has always been of interest to the commercial world, and with the ever increasing demand for accurate indoor localisation and rapid growth of technologies, it has become a quickly developing field. The goal of this thesis was to create a system that could passively monitor WiFi radio waves and use them for localisation while reducing the cost and complexity compared to prohibitively expensive and technical alternatives.

It was discovered that WiFi localisation could be performed with very inexpensive hardware to a reasonable level of accuracy. The ESP8266, an extremely low-cost microcontroller, was used and deployed as a WiFi sniffer, reporting back the signal strengths of nearby devices that are using WiFi to a server. The basestation hardware selected was a cheap single board computer, the Raspberry Pi. Using the Log Path-Loss Model, the captured signal strength of a device could be converted to a rough estimation of distance. Finally, using the data from multiple sensors, trilateration was used to calculated the most likely position of the device.

After optimisations, the final system had an average accuracy of around 1m to 2m based on the size of the room and the density of sensors. It was discovered that a significant portion of the error is most likely due to multipath fading or external sources of 2.4GHz noise, causing reported distances of devices to have large spikes that could occasionally reach meters of inaccuracy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Analysing the movement of people within buildings has always been of interest to building managers, architects, advertisers, and can be life-saving when it comes to safety and evacuation planners. There is an ever increasing demand for accurate indoor localisation, accelerated by the growth of technologies such as augmented reality, social networking, healthcare monitoring and personnel tracking [1,2]. Combined with the increasing popularity of Wi-Fi networks and decreasing hardware costs, Wi-Fi is becoming an attractive basis for localisation systems, with significant research being done in this field for the last 15 years [3,4]. Additionally, using Wi-Fi as the base technology allows for the tracking of most Wi-Fi enabled devices without any additional software or infrastructure.

Current Wi-Fi-based localisation designs are inadequate and depend on precise knowledge of the device and the room's geometry, requiring a significant degree of pre-deployment effort [5]. These designs have accuracies in the range of 6-8m [6] and can rely on the user's device to be actively calculating its location, which is both battery inefficient and requires third party software on the user's device.

The approach taken by this thesis is to passively monitor the Wi-Fi traffic generated by a user's device. As a user's device sends or receives data on a WiFi network, a unique address is attached to that data to identify the sender and recipient devices, normally a phone and a router. A rogue device can monitor nearby radio waves and read both the addresses and strength of that signal. After converting the signal strength to distance, three or more of these sensors combined allows for the calculation of a device's location and with each wireless device having a unique hardware address, identification and tracking can be done between and during visits. The aims of this project are:

1. To develop algorithms for received signal strength indicator (RSSI)-based localisation in rooms or buildings.

2. To optimise positional accuracy and identify the factors that affect it.

3. To propose strategies for how Wi-Fi users or Wi-Fi hardware developers could mitigate user tracking to protect privacy.

Recently, Wi-Fi radio receivers have become small, inconspicuous, and easily concealed. Therefore, WiFi-based tracking and localisation systems are undetectable to those being tracked, and this presents a real privacy concern. As a result, it is important to expand the research done in this area and potentially design methods to counteract these types of tracking systems.

Chapter 2

Literature Review

2.1 WiFi

WiFi is one of the most common methods of wireless communication and with over 7.1 million hotspots, it currently carries more than 60% of the world's Internet traffic [7, 8]. This technology allows electronic devices to connect to a wireless LAN using the 2.4 gigahertz and 5 gigahertz bands. WiFi is used by a large range of devices; these include personal computers, video-game consoles, smartphones, digital cameras, tablet computers and digital audio players [9].

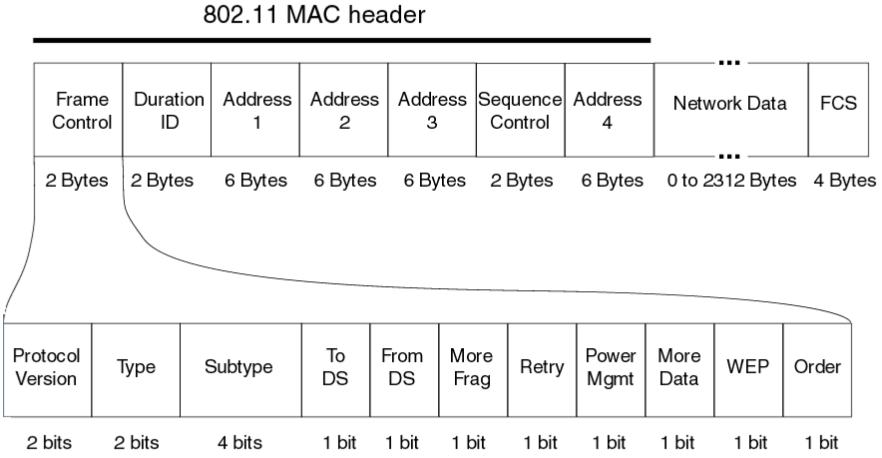


FIGURE 2.1: The 802.11 WiFi packet layout. Source: espressif.com

Figure 2.1 show the layout of a WiFi packet. A MAC address is a unique identifying 48-bit number assigned to hardware with WiFi capabilities and is used to identify which device a packet is destined for. The MAC address is vital to the operation of this thesis and is shown in the Figure by Address 1, 2, 3 and 4.

The "To DS" and "From DS" bits signify what each of the addresses is currently storing. e.g. When "To" and "From" are both set to 1:

- Address 1 = Receiver
- Address 2 = Transmitter
- Address 3 = Destination
- Address 4 = Source

When a wireless network is setup to be encrypted, "Network Data" is the section that becomes encrypted. The header, containing all the MAC address, is completely unencrypted as each device must be able to easily read a packet and determine if it is destined for them. Therefore, even with the highest levels of security placed on a network, the MAC address can still be very easily sniffed out of the air.

As a device uses data across a WiFI network, it generates a large number of packets. These packets are picked up by the hardware designed in this thesis and are then used to locate devices. The more packets that a device generates, the more data points the software has when locating the device and the easier it becomes. However, even when a device is not actively sending data over the network, it will send a keepalive packet at regular intervals to ensure the connection is not lost or forgotten between the two devices. Additionally, when WiFI devices are not connected to a network, they will periodically send out data looking for known previous networks to establish a connection. These features of WiFi allow the designed system to track devices even if they are not using a network. All that is required is that their WiFI is enabled.

2.1.1 RSSI

Received Signal Strength Indication (RSSI) is the measurement of signal strength from a transmitter to a receiver and is the most commonly utilised measurement for localisation. In its raw form, RSSI is measured in dBm. dBm is the power ratio in decibels of measured power compared to one milliwatt and is commonly used to define signal strength in wireless devices. The power in dBm is calculated as

$$x = 10 \log_{10} \frac{P}{1\mathrm{mW}},$$

Where x is the signal in dBm and P is some arbitrary power.

Using a propagation model of Wi-Fi signals and its signal strength, distance can be determined between the two devices [10]. According to the Log-Distance Pathloss Model, the distance is related to the RSSI by

$$RSSI = -10n \log_{10}(\frac{d}{d_0}) + A_0,$$

where d is the distance in meters, n is the signal propagation exponent and A_0 is the referenced RSSI value at d_0. It is common for d_0 to be chosen such that A_0 becomes the signal strength measured at 1 metre.

Here n is a constant that differs from environment to environment, and must be individually measured for each location. For indoor localisation, it is commonly set to 22 [11]. See Table 2.1 for common values of n.

Frequency band	Office area	Commercial area
900 MHz	33	20
1.2 GHz	32	22
1.3 GHz	32	22
1.8 GHz	30	22
2.4 GHz	29	22
4 GHz	28	22
5.2 GHz	31	22
60 GHz	22	17

TABLE 2.1: Calculation of Distance Power Loss Coefficient. Data from Ref. [11].

2.1.2 Multipath propagation

Multipath propagation is the phenomenon where radio signals bounce and reflect before reaching the receiving antenna. This interference includes constructive, destructive and phase shifting of the signal, as seen in Figure 2.2. Multipath propagation represents a challenge for indoor localisation as it is an enclosed space with many non-static objects, this creates an environment that makes separating the noise from the signal a difficult task. However, over the years, many techniques have been built to minimise the errors caused by this phenomenon. These techniques are discussed in sections 2.3.2 and 2.3.3.

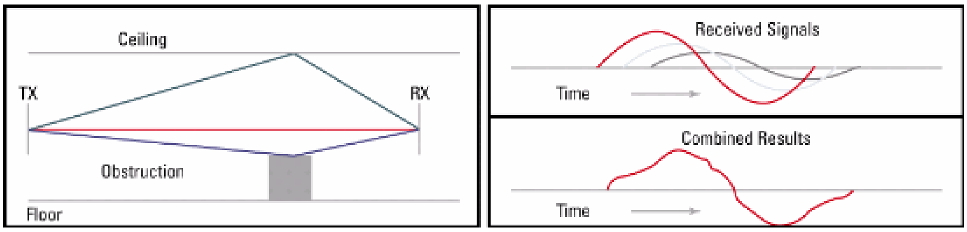


FIGURE 2.2: The effects of Multipath Propagation. Figure from Ref. [12].

2.2 Localisation Techniques

Over the years, many techniques have been created to satisfy the ever growing business interest of indoor localisation. In today's standard, high accuracy is considered to be 1 to 5 metres, medium accuracy is 6 to 10 metres, and low accuracy is over 11 metres [13]. A variety of localisation techniques have been selected and discussed below.

2.2.1 RSSI and Trilateration

The simplest form of localisation is the combination of the RSSI Log-Distance Pathloss Model and Trilateration. The approach has multiple sensors placed around a room that record the signal strength of a device. Using Trilateration, several distance measurements can be combined to find a location. As it is known that if there is an intersection of three or more circles, then this provides enough information to pinpoint a location as illustrated in

Figure 2.3. If there is no perfect intersection a common modification is to add a centre of mass calculation. This approach consists of calculating and selecting the most probable location and is known as Multilateration [14].

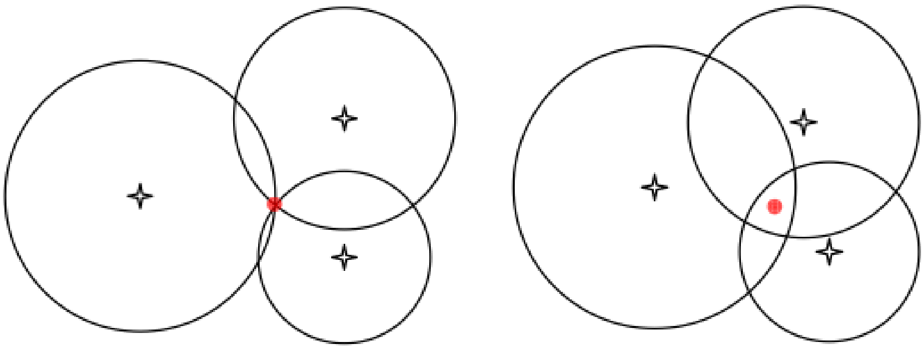


FIGURE 2.3: An example of Trilateration (left) and Multilateration (right). Source: rvmiller.com

2.2.2 Fingerprinting

Fingerprinting is the recording of signal strengths from various sensors along with known location data while in an "Offline" phase. This information can be either deterministic or probabilistic [4, 15]. During "Online" tracking, current measurements are compared with the database and the closest match is returned [16].

Fingerprinting is the most common technique for localisation [17]. Although it has among the best accuracy at around 2 metres [18] and is one of the easiest methods to implement, the main disadvantage of this technique is a significant amount of man-hours required to calibrate an area. Additionally, this calibration must be completed every time there is a change in environment [5, 19]. However, it is predicted that having a high density of sensors will help overcome this obstacle as the majority of sensors will show the same reading after smaller changes in the environment.

Fingerprinting is typically combined with low range sensors located in various high-density layouts. While this technique requires a significant number of calibration points, it can be one of the most accurate positioning setups. Table 2.2 shows the results from a 10x10m room with four access points, one in each corner of the room. This method allows for easy and methodical testing as the accuracy of this approach can be improved by increasing the number of access points and calibration points.

Number of calibration points	**Average Error (m)**
36	1.92
121	1.90
256	1.83

TABLE 2.2: Grid Concept. Calibrations vs. Accuracy. Table from Ref. [20].

2.2.3 Choke Points

Choke Points is a technique similar to fingerprinting in which sensors are located in high-interest positions. However, this usually amounts to only one sensor per room or hallway. Unlike RSSI techniques, these commonly do not report signal strength or distances; rather a device is said to be in the same location as the sensor [15]. The accuracy of this technique is poor (6-10m), and while it will locate the room a target is in; it cannot determine if the target is at the front or back of that room. Choke Points is mostly used in locations with strict budgets as the main advantage of this technique is that it has very little hardware requirements, due to most locations already having WiFi networks that can be reused and no needed initial or frequent calibration.

2.2.4 Angle of Arrival

Angle of arrival is the measurement of the angle in which a signal hits a receiver and is accomplished by measuring the different times of arrival to two different antennas. It can also be achieved more simply with the use of directional antennas. However, this technique relies on proper placement of the antennas [21]. Recent studies have achieved an average accuracy of 6m using this technique, placing it medium accuracy category [22]. This technique has the advantage of requiring little to no calibration as the setup relies on a mathematical model. However, calibration can be implemented to increase the accuracy of an existing system. This system requires very advanced hardware and as such is prohibitively expensive for most companies to setup making this a non-ideal solution for indoor localisation.

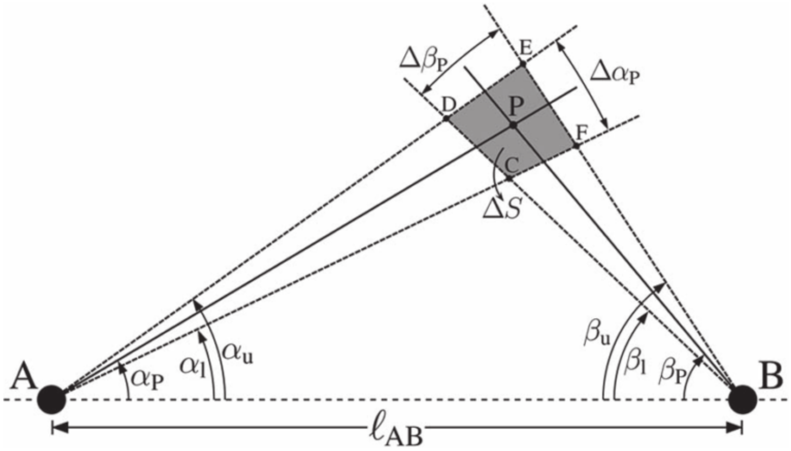


FIGURE 2.4: An example of angle of arrival and the inaccuracies produced by the angular errors. Figure from Ref. [22].

2.2.5 Time of Arrival

Time of arrival is the measurement of time it takes for a signal to propagate from the transmitter to receiver. Because the rate of signal propagation is constant and known, the travel time can be used to calculate distances. This distance is then combined with Trilateration as discussed in Section 2.2.1.

Time of Arrival can have accuracies in the range of 1-3m [23]. However, both time of arrival and angle of arrival often suffer from multipath conditions when used indoors and require complicated synchronisation mechanisms [24,25].

Time of arrival is usually performed with similar hardware to angle of arrival and as such is also prohibitively expensive to setup.

2.3 Mathematical Filtering

2.3.1 Least Squares

Least Squares is a regression analysis technique used to approximate a solution. The technique minimises the difference between an observed value and the fitted value provided by a model as seen in Figure 2.5.

This technique is essential for indoor localisation as not all sensors will report exact or matching data, and is the core calculation of multilateration allowing it to find the "most correct" position.

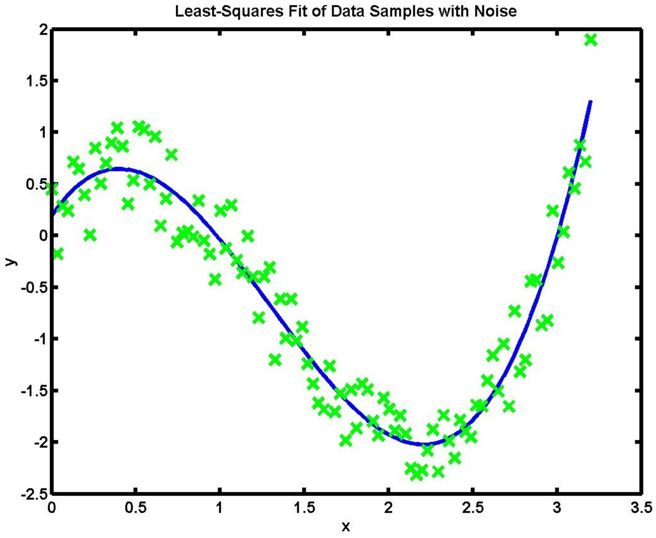


FIGURE 2.5: Least Squares being applied to find a line of best fit. Source: dmpeli.math.mcmaster.ca

2.3.2 Kalman Filter

Kalman filtering is a method of filtering out the noise and other inaccuracies from a set of data. It is especially prevalent in Indoor and WiFi localisation. The algorithm uses past measurements and produces estimates that tend to be more accurate [14].

$$X_t = A_t X_{t-1} + B_t U_t + \epsilon_t$$

The curent state X_t is a combination of X_{t-1}, a control input U and noise ϵ. Here A is the state transition model and is applied to X_{t-1}, B is the control input model and is applied to the control vector U and ϵ_t is the process noise.

The algorithm works in two parts. The first part is known as the "prediction step", produces estimates of the current state variables, along with their uncertainties. Once the next measurement is obtained, the estimates are used to update a weighted average, with more weight given to estimates with higher accuracy.

This technique is one the most successful when it comes to eliminating noise in radio signals and is the most used method when it comes to RSSI-based localisation. This technique can be easily implemented via libraries such as "Pykalman" or "FKF".

2.3.3 Exponentially Weighted Moving Average

An exponentially weighted moving average is a moving average that applies exponentially decreasing weighting on old values. In the formula below, X is the current value of the moving average, Y is the newest value to be added to the average, and α is an integer between 1 and 0 which determines the weight placed on old values. When implementing this method, it is important to choose an α that is not too small and thus does not average or remove outliers, or a value that is too large and causes the average to take a considerable time to change, decreasing the responsiveness of the system.

$$X_t = Y_t$$

$$X_t = \alpha Y_t + (1 - \alpha) X_t$$

This technique is applied to remove outliers from signal strength measurements caused by multipath propagation.

2.4 Privacy

While WiFi hotspots simplify our online activities, WiFi is less secure than a wired connection such as Ethernet, as intruders do not require a physical connection and with the rise of WiFi enabled devices, these networks are more dangerous than ever. Most users are not aware of the inherent threats of using a public network, such as; identity theft, malware, and credit card, banking or log-in details being stolen. These public networks are almost always unencrypted, meaning anyone with cheap and easily available software can listen to anything being sent over the network [8]. While encryption may help to hide the websites that a device is visiting, devices are still vulnerable to passive tracking technologies.

This passive tracking technology is already used in various places today e.g. advertisers and retailers in shopping centres for tracking customer shopping behaviours, analytics startups, or even in hospitals that track patients and visitors. An example of this technology is FootPath, developed by Path Intelligence. It uses devices located around a shopping centre to help retailers better understand customer browsing and purchasing behaviour.

While it is legal to track users, it is only true when the company is not storing personally identifiable information. Unfortunately too often anonymity is broken, releasing personal information into the wrong hands. This is especially true as the majority of the companies currently developing passive tracking technologies are small startups that do not have the manpower required to fully secure a system. Additionally, it is only legal if a user can opt out of the system. However, most tracking setups rarely, if ever, notify customers they are being tracked. If a customer does not know the system exists, how can they possible opt out of it?

The solution is relatively simple. In theory, as long as you have your WiFi turned off or have your phone set to flight mode (depending on the tracking technology used), the device will no longer send packets that can be tracked. However, the inconvenience of having to turn off your WiFi or enter flight mode every time you leave your house is enough to deter most people.

2.5 Commercial Systems

For passive systems Path Intelligence created the FootPath. Rather than using WiFi or Bluetooth, FootPath uses the regular pings to cellular towers that each phone makes to track a device. However, its accuracy is only to the level of knowing what store a customer has visited or how long they spend in a shopping centre.

Another popular tracking system is iBeacons. However, it is not a passive tracking setup and requires the user to download an app to find their location. It currently costs $40AUD per a device and requires a device every 1.5 metres for their highest accuracy setting of 50cm [26]. Additionally, the batteries in the iBeacon devices only last about a month and incur hefty maintenance fees [26]. With the ever increasing demand for indoor localisation, there is a considerable need for a cheap and effortless system, particularly those that can create a high density of sensors.

2.6 Conclusion

There are numerous techniques for obtaining indoor locations, and they vary in both cost and complexity. Of these techniques, the simple RSSI and trilateration setup is one of the most robust and easy to setup methods and is the go-to system for current setups. However, there is a problem with existing systems. They are expensive. Furthermore, as these methods of WiFi localisation are completely passive, it is undetectable by those being tracked. Therefore, of the networks that are currently deployed, there is a serious security and privacy concern to those within the vicinity.

Chapter 3

Methodology

3.1 Implementation

The previous chapters identified a need for an accurate indoor localization system designed for and used by the commercial world. Therefore the design choices of the system are centred around the absolute lowest production cost and easiest availability, allowing for widespread deployment and uptake of the technology. Below describes how the passive tracking system was designed.

3.1.1 Algorithm Selection

The simple RSSI and trilateration setup was chosen as it is one of the easiest and most effective ways to setup a localisation network while also requiring the simplest level of hardware. This type of system has many benefits, the most important being that it is cheap and allows for quick and fluid testing. The changing of the grid and other variables follows a straightforward and logical pattern and as such this method is extremely easy to deploy in almost any building as the location of the sensors is not restricted and can be scaled to any size room or building.

Other methods like angle of arrival and time of arrival are too complicated, requiring advanced hardware. As such these systems do not promote large numbers of sensors and are prohibitively expensive to both test and deploy.

Although using the simplest method of localisation allows for the possibility of substantial inaccuracies in tracking, it was primarily chosen due to its low cost. The extremely low price of the hardware allows for a very dense network of sensors reducing this problem.

The formula implemented for the RSSI design was the Log-Distance Pathloss Model, the distance is related to the RSSI by

$$RSSI = -10n\log_{10}(\frac{d}{d_0}) + A_0,$$

where d is the distance in meters, n is the signal propagation exponent and A_0 is the referenced RSSI value at d_0. It is common for d_0 to be chosen such that A_0 becomes the signal strength measured at 1 metre.

Additionally, using Least Squares with Log-Distance Pathloss Model, which is a regression analysis to approximate a solution for overdetermined systems, allows for a location to found for each of the devices. The same least squares formula is used to determine the optimal coefficients during a training phase, which are then used during runtime.

3.1.2 Base Station Hardware

While any computer could have been used for the base station, a small Linux computer like the Raspberry Pi is the perfect solution for this setup. Not only do these platforms allow for much easier development, but they are extremely cheap and easily available. These devices are also quite capable when it comes to the number crunching of the algorithms.

	BeagleBone	**Raspberry Pi**
Price	$45	$35
CPU	2x 1GHz	4x 1.2GHz
RAM	512MB	1GB
Storage	2GB on-board + SD card	SD card only
Video	1 Micro-HDMI	1 HDMI
Power Draw	200-400mA	150-350mA
GPIO	65 Pins	40 Pins
Peripherals	1 USB + 10/100 Mbps Ethernet	2 USB + 10/100 Mbps Ethernet

TABLE 3.1: Comparison of the two most popular Linux single board computers.

The Raspberry Pi is one of the most popular development platforms. Its low price, small form factor and extremely well-supported environments for developers make it an attractive platform for most hobbyist. The Raspberry Pi can run a multitude of operating systems and supports all of the most popular software packages, allowing for fast and easy implementation of the server.

Like the Raspberry Pi, the BeagleBone Black is a low-cost, community-supported development platform for developers and hobbyists. While a relative newcomer to the world of easy to use microprocessor breakouts, it has excellent performance over the competition and may effective as a base station.

However, the Raspberry Pi was selected as base station due to its availability, since it can be easily borrowed from the University, as well as experience in using and setting them up.

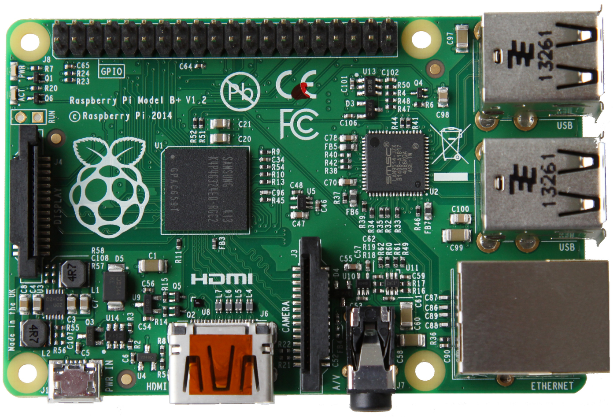


FIGURE 3.1: The Raspberry Pi. Figure supplied by www.raspberrypi.org

3.1.3 Wi-Fi Sensor Hardware Selection

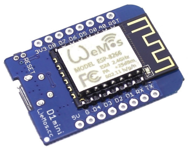


FIGURE 3.2: The WeMoS D1 Mini. A small form factor microcontroller with WiFi capabilities. Figure supplied by www.cnx-software.com

While a Raspberry Pi and WiFi dongle could perform this job, the solution ends up larger, more power intensive and expensive compared to a microcontroller solution. Since the tracking setup only requires RSSI, a very simple and cheap microcontroller, the ESP8266, was used to accomplish this task easily.

Name	Memory	GPIO	Price	CPU	RAM
NodeMCU V3	4MB	10	$11	80MHZ/160MHz	20KB
ESPDuino	2MB	44	$10	80MHZ/160MHz	160KB
WeMoS	4MB	11	$4	80MHZ/160MHZ	160KB
WiFiMCU	2MB	17	$10	100MHz	128KB

TABLE 3.2: Selection of possible ESP8266 Devices. Green indicates the best in that category.

The ESP8266 development boards were selected for the easy availability, low price, and programming flexibility. Creating a cheaper solution is important as it allows for more sensors to be deployed out in the field, potentially increasing accuracy tracking results. Additionally, the ESP8266 has attracted many 'hackers', and as such, it has an extensive knowledge base on the topic being researched. Table 3.2 shows the comparison of common development platforms.

The WeMoS D1 Mini was chosen as the ESP8266 development platform for the sensors as not only is it the cheapest, it wins in all specifications, except GPIO, which is not utilised in this thesis. Additionally, the microcontroller was tested before usage in the thesis to prove that it was capable of sniffing and has enough processing power to handle the task.

3.1.4 Final Hardware Selection

From the hardware discussed above, Table 3.3 shows the final selection of hardware that was used in the Thesis.

Name	Price (AUD)	Qty	Total
WeMoS D1 Mini	$ 5.30	15	$ 79.50
Lipo Battery	$ 2.22	15	$ 33.30
10 Pairs 100mm JST Connector	$ 2.64	2	$ 5.28
Raspberry Pi 3 Model B	$ 56.00	1	$ 56.00
16GB MicroSD	$ 20.00	1	$ 20.00
2.5Amp 5.1V Power Supply	$ 13.00	1	$ 13.00
		Total	$207.08

TABLE 3.3: The Bill of Materials for final selection of hardware.

3.1.5 Wireless Sensor Software and Data Transport

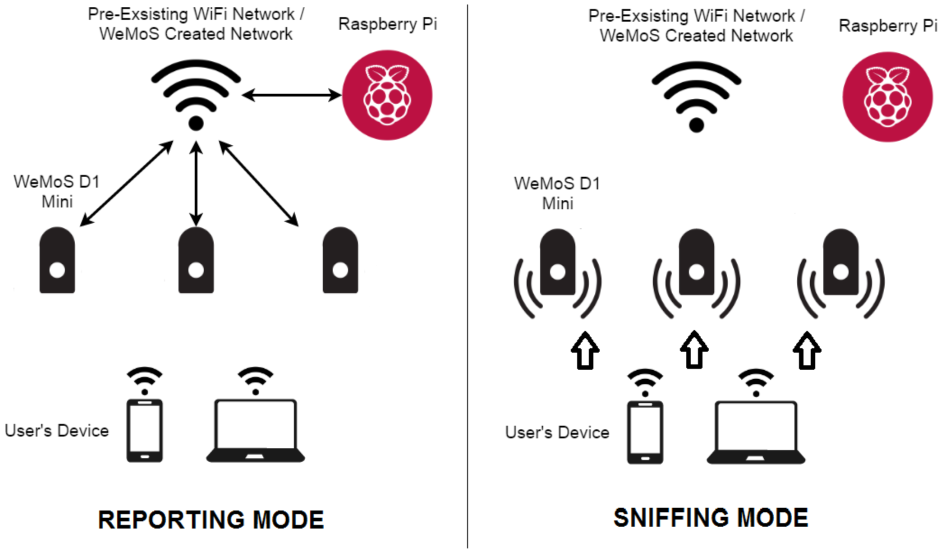


FIGURE 3.3: Layout of WeMoS Network. Shows both stages. Sniffing where the WeMoS devices will scan the airwaves for information. Reporting, where each WeMoS uses an existing WiFi network to report back to the base station.

A simple approach of using a pre-existing network and a base station to receive the data was chosen. Figure 3.3 shows the current setup for the WeMoS devices. Each WeMoS device scans and sniffs the area around it for wifi packets, and at a regular interval will use a pre-existing wifi network to send data back to the main server discussed in Section 3.1.2.

While the first edition of the sensors used a JSON based techniques for sending data, the large memory requirements of it meant that each device could only monitor and locate a limited number of devices. The final edition uses Google's Protobuf technique for packing data; this technique has increased the amount of data that can be stored and sent by almost ten fold. This technique also takes significantly less time to process and prepare for sending, meaning the data is fresher when it reaches the server. The moving average and Protobuf techniques are necessary as if a WeMoS device goes over 90% memory usage, it becomes unreliable, failing to connect to networks.

The WeMoS was also discovered to be able to monitor WiFi management packets. These 112-byte packets transmit from most devices around every 4 seconds, allowing this system to observe all devices even when they are not actively using data on the network or are searching for a network. This creates a serious ethics issue as even if a device does not connect to your network, you are still able to automatically opt them in and track them.

Since these devices use Lithium polymer batteries, it is important that each device can monitor and manage each of their battery levels. Currently, the devices will report back the current voltage so that a user may have a live view of how each device is going and if it is going to run out soon. The devices will also automatically shut themselves down when the battery levels get to around 10% remaining; at this point they enter a deep sleep using only 20nA of power, giving the user around one month to unplug the device before the battery is permanently damaged.

Each of the WeMoS devices is designed to monitor how long they take to connect to the network and send data, allowing for all of the information sent to be automatically adjusted, giving the user within 1 second of accuracy. Examples of this timing and be seen in Figures 3.7 and 3.8.
The Sensors have a defined set of flashing lights to communicate what stage the device is in, allowing for easy synchronisation attempts and debugging.

- Fast Flashing Light -> The device is currently attempting to connect or find the WiFi network
- Light Off -> Device is currently searching for the server that manages each of the devices
- Light Solid -> Connected to the server and waiting for a command
- Light Blinking (3 Seconds) -> The device is current running in sniffing mode

Timing Trade-offs

The designed system uses the same WiFi device to sniff data and report back data. Since the wireless devices can only be in a single mode, either sniffing or reporting, there is a trade-off between percentage uptime of the system and the responsiveness.

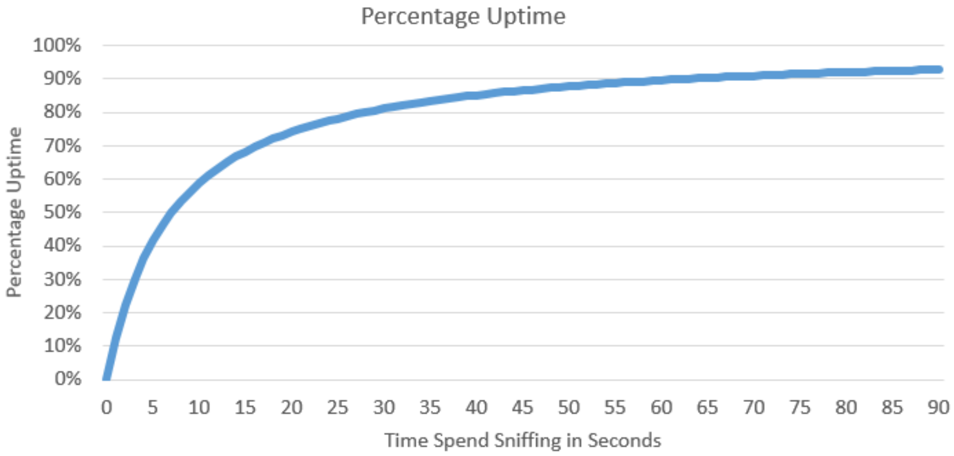


FIGURE 3.4: The percentage of time a WeMoS is sniffing base on the response selected for the system.

Figure 3.4, shows the trade-off between time spent sniffing vs. time spent uploading and the delay that comes with that trade off. As each of the devices takes around 4 to 7 seconds to post data, having the WeMoS sniff for 7 seconds means it is only able to capture positions half of the time. The tradeoff here is that while a system may be able to track 90% of the time, there is a 60-second delay. However, having a delay of 10 seconds allows for a significantly faster update, but the sensors will miss around 40% of all packets.

Figure 3.5 shows the trade-off that is made between the response time of the system, the resolution of the data and the number of devices that can be supported at once.

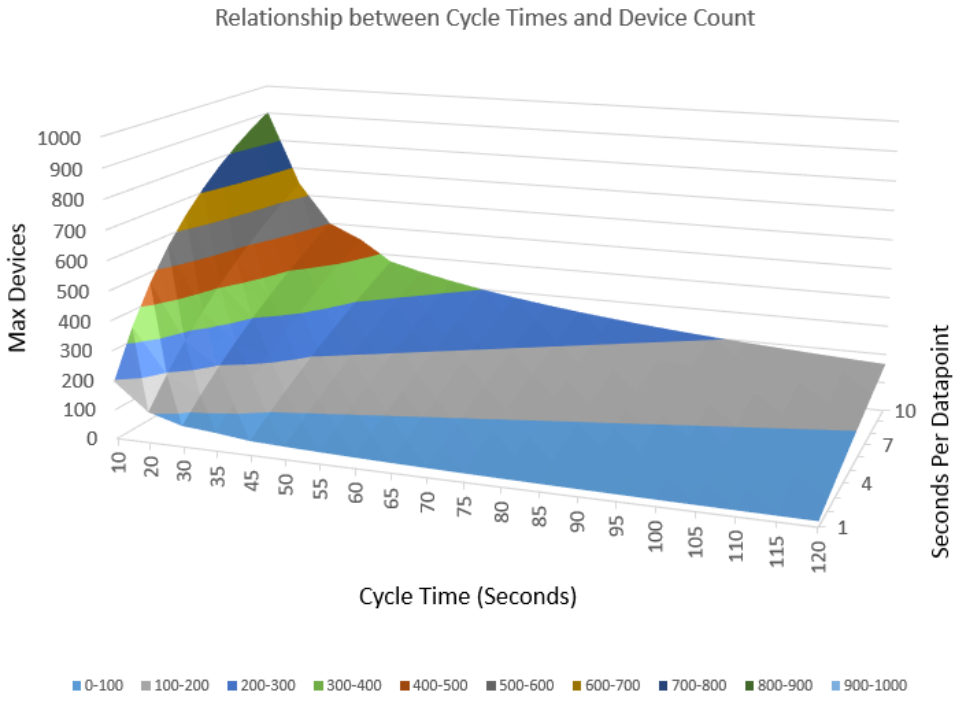


FIGURE 3.5: The trade-off that is made between the response time of the system, the resolution of the data and the number of devices that can be supported at once.

As the main goal of the system was to improve its accuracy, the response time could be set very high to ensure almost all the data was sniffed. The final number chosen was 55 seconds, giving 90% sniffing uptime. The data point frequency was set to 1 data point per second, giving the mathematics a significant amount of data to work with.

As such each of the WeMoS devices track the last 55 seconds worth of data. Over that period the data is input to a moving average for each device. At the end of the 1 second period, that data is stored as the actual signal level at that second. Not only does that dramatically reduce the amount of data that must be sent, but also produces much more accurate distance measurements.

3.1.6 Database Software Setup

Multiple database setups were tested to find the most appropriate fit.

Phant is the database software used by Sparkfun. It is extremely easy to use and very popular. However submitting to the Phant software takes up to one second per entry due to having to restart the connection each time. This creates submission times up to minutes when there is a large amount of information to be sent from a device and as such is not suited for the job.

MariaDB is an SQL database. SQL databases are one the most common database types and could be a good choice for this thesis. However, SQL databases can be extremely complex and hard to use. For storing such simple data from these sensors, the additional complexity of managing a database is unreasonable. The learning curve and development time for this software would be inappropriate when compared to the fast development and history of knowledge when it comes to a relatively simple server written in Python.

MongoDB is a NonSQL based database. It is extremely easy to use. However, it can become extremely messy as the database gets larger and larger. There is also the requirement of an additional process constantly running, increasing the effort required to setup and run this kind of database.

The last option was to code the database by hand. While this requires development time, developing custom software allows for exactly the right features and setup that is desired for the project.

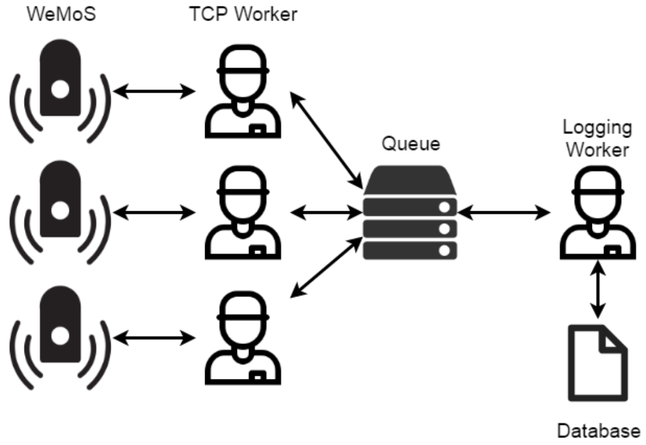
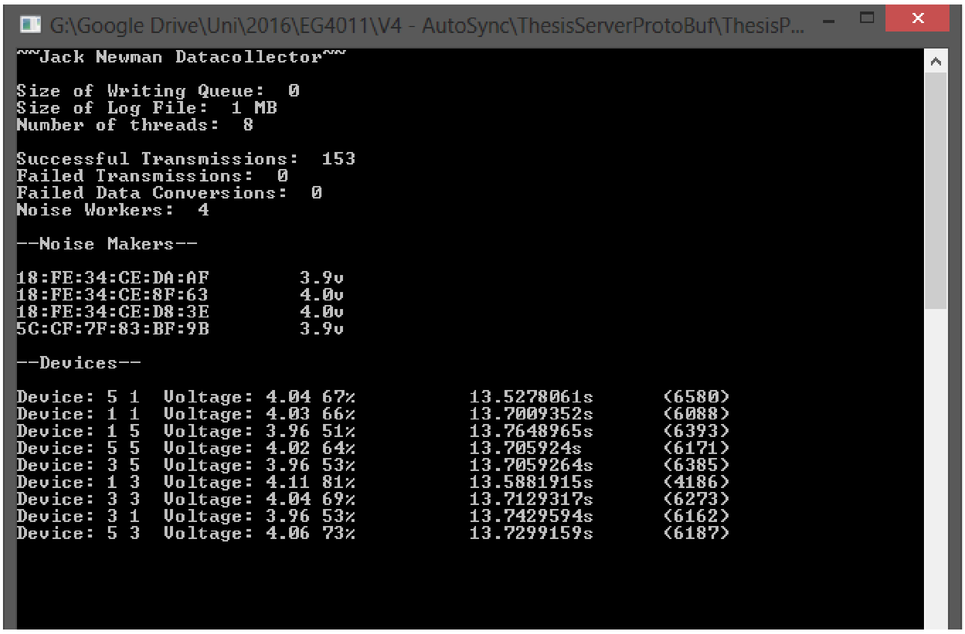


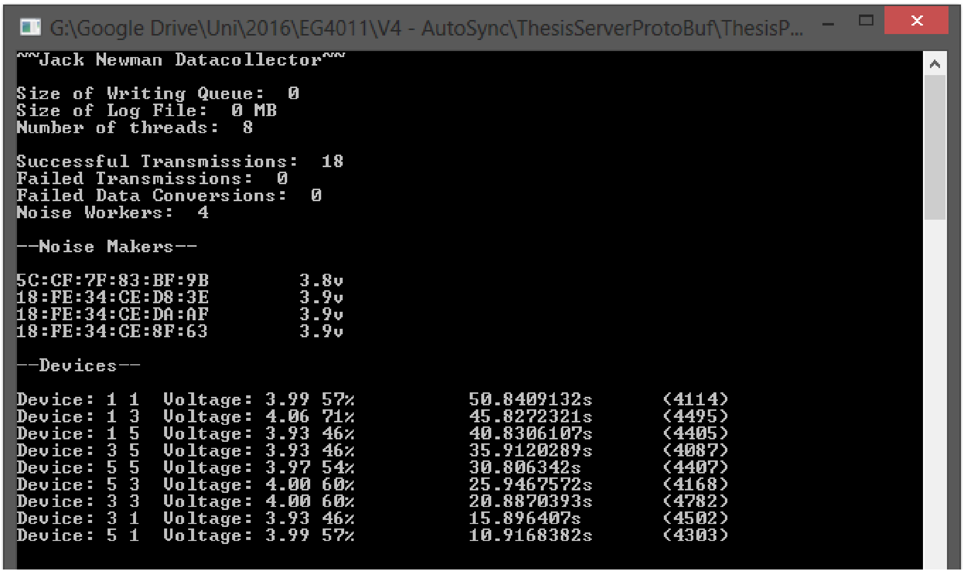
FIGURE 3.6: Layout of the Database Workers. A thread for communication with each WeMoS device, and a thread to write the data to disk.

Figure 3.6 displays the setup for the server. There is an individual worker for each WeMoS device allowing for many devices to report data at once, with one worker whose job is to write the combined data to a log file.


G:\Google Drive\Uni\2016\EG4011\V4 - AutoSync\ThesisServerProtoBuf\ThesisP...

```
~~Jack Newman Datacollector~~

Size of Writing Queue:  0
Size of Log File:  1 MB
Number of threads:  8

Successful Transmissions:  153
Failed Transmissions:  0
Failed Data Conversions:  0
Noise Workers:  4

--Noise Makers--

18:FE:34:CE:DA:AF        3.9v
18:FE:34:CE:8F:63        4.0v
18:FE:34:CE:D8:3E        4.0v
5C:CF:7F:83:BF:9B        3.9v

--Devices--

Device: 5 1  Voltage: 4.04 67%          13.5278061s      (6580)
Device: 1 1  Voltage: 4.03 66%          13.7009352s      (6088)
Device: 1 5  Voltage: 3.96 51%          13.7648965s      (6393)
Device: 5 5  Voltage: 4.02 64%          13.705924s       (6171)
Device: 3 5  Voltage: 3.96 53%          13.7059264s      (6385)
Device: 1 3  Voltage: 4.11 81%          13.5881915s      (4186)
Device: 3 3  Voltage: 4.04 69%          13.7129317s      (6273)
Device: 3 1  Voltage: 3.96 53%          13.7429594s      (6162)
Device: 5 3  Voltage: 4.06 73%          13.7299159s      (6187)
```

FIGURE 3.7: The new Golang server, displaying a test of the sensor network with all the sensors in sync.


G:\Google Drive\Uni\2016\EG4011\V4 - AutoSync\ThesisServerProtoBuf\ThesisP...

```
~~Jack Newman Datacollector~~

Size of Writing Queue:  0
Size of Log File:  0 MB
Number of threads:  8

Successful Transmissions:  18
Failed Transmissions:  0
Failed Data Conversions:  0
Noise Workers:  4

--Noise Makers--

5C:CF:7F:83:BF:9B        3.8v
18:FE:34:CE:D8:3E        3.9v
18:FE:34:CE:DA:AF        3.9v
18:FE:34:CE:8F:63        3.9v

--Devices--

Device: 1 1  Voltage: 3.99 57%          50.8409132s      (4114)
Device: 1 3  Voltage: 4.06 71%          45.8272321s      (4495)
Device: 1 5  Voltage: 3.93 46%          40.8306107s      (4405)
Device: 3 5  Voltage: 3.93 46%          35.9120289s      (4087)
Device: 5 5  Voltage: 3.97 54%          30.806342s       (4407)
Device: 5 3  Voltage: 4.00 60%          25.9467572s      (4168)
Device: 3 3  Voltage: 4.00 60%          20.8870393s      (4782)
Device: 3 1  Voltage: 3.93 46%          15.896407s       (4502)
Device: 5 1  Voltage: 3.99 57%          10.9168382s      (4303)
```

FIGURE 3.8: The new Golang server, displaying a test of the sensor network with all the sensors in out of sync to reduce load on the server and WiFi.

The original database software was coded in Python; however, the current version has been programmed in Google's new language Golang. Golang is a language orientated around server and multi-threaded applications, making it a perfect fit for the Thesis's requirements.

Through the use of a test client program, the server software was benchmarked to determine how fast and how many devices it could support, allowing for easy and quick testing of different software setups. Golang improved the CPU of the server and the number of devices that could be handled simultaneously, reaching upwards of 3000 devices submitting every five seconds on a non-server based OS, limited only by the usage of all the available ports.

Since the WeMoS devices were changed to Protobuf, the server was also changed to used Protobuf, significantly reducing the CPU load on the server as Protobuf can be Unmarshaled 100's of times faster.

An additional feature was added to the server that allows for the synchronisation of the sniffer devices. Devices connect to the server on bootup and then wait for the start command before actually running any of the sniffing software. This allows for each of the WeMoS devices to post their data in synchronisation, freeing up the airwaves from unnecessary noise. Note that it is also possible to equally split up the time at which each device submits to reduce the peak load on the WiFi network and the server.

Figures 3.7 and 3.8 show the new server in action with the sensor network both in sync or spread out to reduce the load.

3.1.7 Calculations

Most of the calculations for the localisations are performed in a language called 'R'. R is a programming language focused primarily on statistical computing and is widely used for its data mining capabilities. While Python could have been utilised as the base language for this section, R is a better alternative as it natively supports many of the mathematical operators required for localisation, and hence significantly reduces development time of the software.

To improve the accuracy of the system, either a moving average or kalman filter needed to be added to the system. Due to the very high data rates it would be inappropriate to send all of this data to the server. As such it is required that the averaging take place on the actual devices, and so the simplest and least performance impacting design of moving average was used.

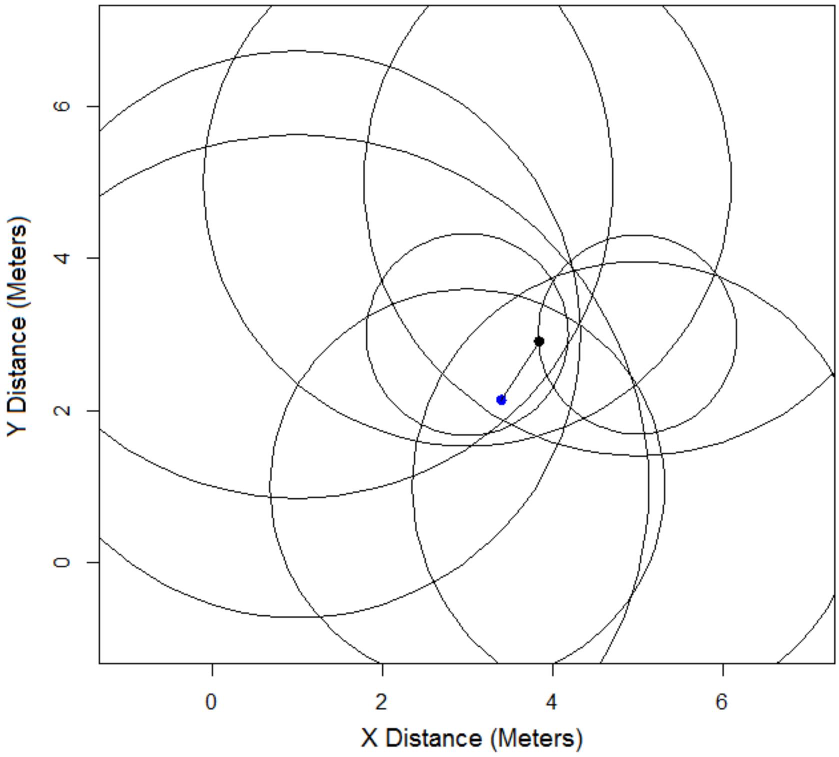


FIGURE 3.9: The calculation of Trilateration in R. Blue showing the correct location of the device, black being the calculated location.

3.2 Testing

3.2.1 Performance Metrics

These metrics were used to assess the performance of the hardware and algorithms. In particular, for Accuracy, it was important to consider the mean and standard deviation of the measurements.

- Accuracy - The distance between calculated position and true position.
- Responsiveness - How quickly is the device's location updated?
- Coverage - How easy is it to add additional areas?
- Adaptiveness - The ability to handle changes in the environment such as the movement of furniture
- Scalability - How easy is it to expand the system?
- Cost and Complexity - Cost of deployment, infrastructure and labour.

3.2.2 Room Layout

The low cost of the hardware allows for these localisation techniques to be performed with a high density of sensors. For the first test, the sensors were placed in a 3x3 grid on the roof shown in Figure 3.10. In this trial the sensors were all set to submit at the same time, reducing the wireless noise in the room. The second test was identical except the noise makers were moved to different locations.

For the third test, the noise makers were moved once again, however, this time, the sensors we setup to submit 5 seconds apart from each other to place an equal load on the WiFi network and the server. This also allows for any changes in accuracy to be measured from having extra noise in the room.

For all of the tests, the number of sensors in the room was identical, however, in the database selected sensor can be removed to simulate doing the tests with fewer sensors. This allows for recording and measuring the resulting change in performance without a constantly changing the physical setup. This technique provides a way to model the density of sensors and the resulting accuracy.

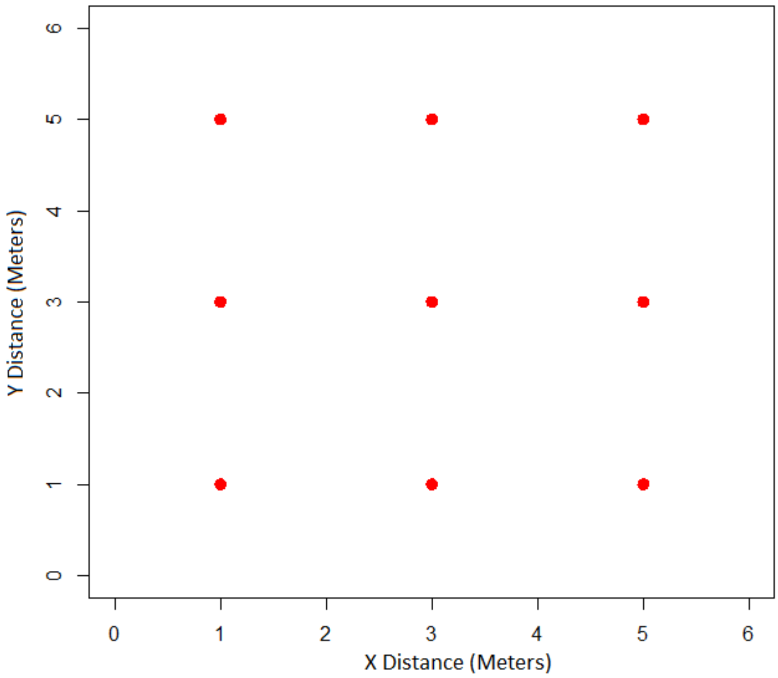


FIGURE 3.10: Example of the 3x3 grid layout of the sensors in a single room.

3.2.3 Testing and Calibration

To allow for easy and quick testing, certain WeMoS devices can be setup as "Noise Makers". It is the task of these devices to connect back to the main server and send a constant stream of data, flooding the airwaves with packets that the WeMoS sensors can then pick-up and use to locate devices. The current version of noise makers are set to produce around 16 packets of data per second. However, this is completely configurable.

To make the life of the user easier, Noise Makers will spit out their MAC on boot over serial and also over the network to the database server to make it easier to keep track of them and record them down. The noise makers also have the ability to monitor their battery. However, they do not report these numbers back to the server. Rather the server will notice that they have died naturally as the constant stream of data will stop.

The Noise Makers also have their own set of flashing light combinations to signify what stage they are currently in.

- Solid Light -> Device is currently booting
- Flashing Slowing -> Device is currently trying to find the database server
- Flashing Quickly -> The device is currently flooding the network with data

3.2.4 Security and Privacy

The two main techniques used to try and defeat this system were modifying the signal strength and changing the MAC address of the device.

Changing the signal strength of the device was not tested as it directly impacts the usefulness of a WiFi system. Reducing signal strength is likely to interrupt a user's session and as such is unlikely to be used in practice.

The second method for defeating the monitoring system is to change a device's MAC address altogether. This will cause the user to appear as an entirely new person each time this happens, effectively splitting the user up into multiple people. However, since the tracking system only works with a pre-existing WiFi network, an attacker would already have access to the network, if a certain device was to disconnect from the WiFI and then a new device was to connect a few seconds later and then immediately re-establish all of the same connections, it becomes rather easy to connect the dots and continue to track a user.

Chapter 4

Discussion

4.1 Testing

The first tests were conducted with a 3x3 grid of sensors on a roof as shown in Figure 4.1.

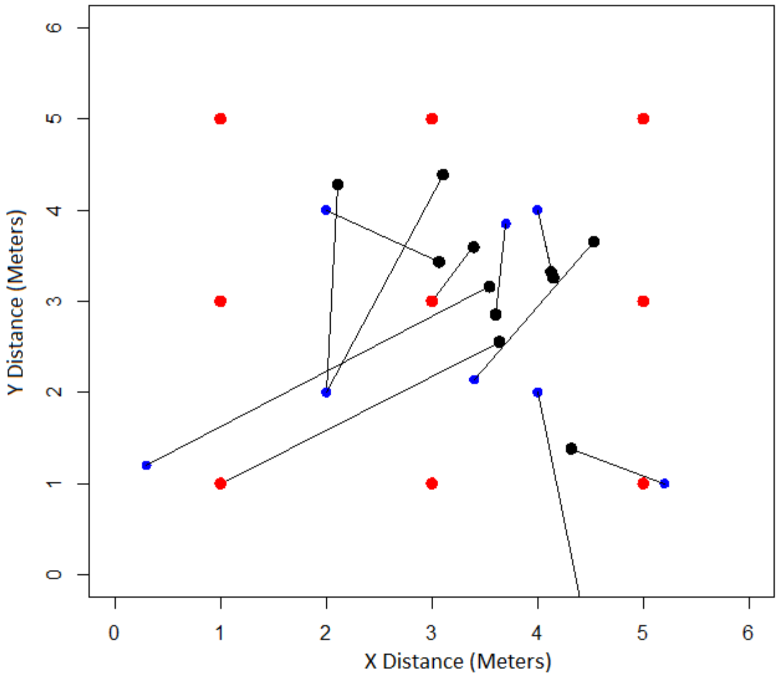


FIGURE 4.1: The first three tests combined and showing the calculated locations after taking roof height into consideration. Red dots are sensors, blue dots are the known location of a device, black dots are calculated locations.

Excel was used to manually calculate the correct results for each of the sensors. However, it was discovered that the height of the room must be taken into account for the calculations to return valid results. Otherwise, all devices are located in the dead centre of the room. An optimal N value was found around 5, giving an accuracy of 1.4 metres.

The system was designed so that the sensors would connect back to the base station and then idle, waiting for a start command. Syncing the sensors so all of them would sniff and then report back data at the same time. The first two tests made use of this feature to reduce any wireless noise that could potentially affect the results. However, the third test had each of the transmitters split apart by 5 seconds to both minimise the load on the server and WiFi network, and to discover if there was any noticeable difference in accuracy. Examples of this feature in action can be seen in Figure 3.7 showing all sensors in sync and Figure 3.8 indicating the 5-second gap between devices. The average accuracy of the first two tests after taking into account 3D space and using Excel to optimise mathematical constants was around 1.35 metres. However, the accuracy of the third test by itself had an accuracy of around 1.12 metres.

4.2 Improving Accuracy

Since the initial results were not as accurate as hoped, a few tests were run to find what part of the calculations was wrong. The first test was an RSSI vs. Distance test to show the noise in movement and to see how badly multipath fading affected the tests. Multiple devices were tested and plotted, with a goal that each of the lines perfectly overlaps.

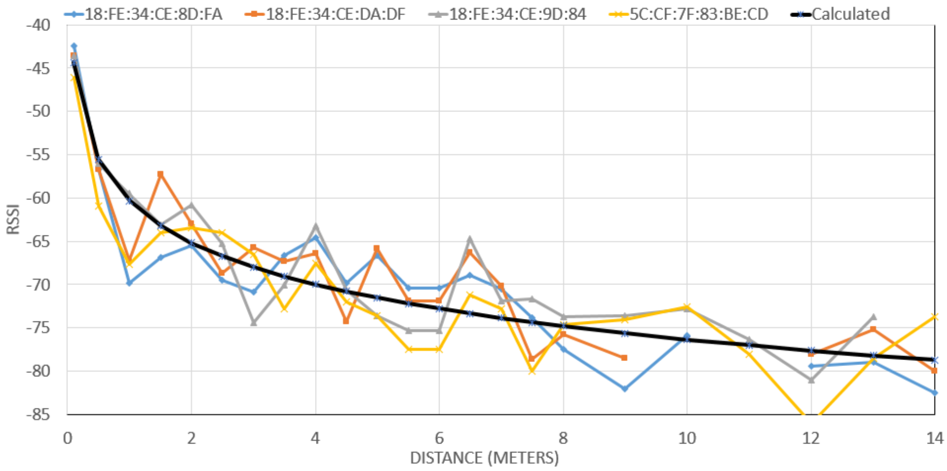


FIGURE 4.2: RSSI vs Distance. Multipath fading causing random large dips in signal strength.

Figure 4.2 shows measured distance vs. RSSI. The test had the sensors measuring data for 60 seconds, which was then averaged to determine the final RSSI for that distance, note there are some missing values where the signal was too weak for the WeMoS to consider it valid. The graph shows that each of the devices does somewhat follow the expected line, however, multipath fading is most likely the cause of the random large spikes. The test tells us that the propagation equation is most likely not wrong, and it is an unavoidable error due to multipath fading. According to Figure 4.2 at the 6.5m mark, the reported RSSI for most devices shot up to around -65dB. This causes an error of almost five metres in the final calculations. Additionally, since RSSI is reported back by the devices as an integer rather than a decimal, even if the report RSSI is the correct, the resolution of the distance is quite large, and the best case is that distance is reported within a 20cm chunk.

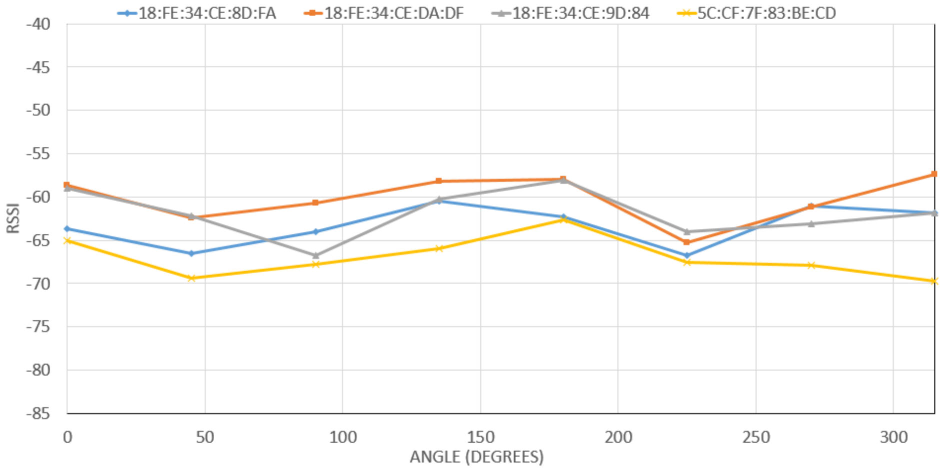


FIGURE 4.3: RSSI vs Angle. The changes in RSSI as a device is rotated 360 degrees.

Figure 4.3 shows the change is RSSI as a device is rotated. The test was performed with each of the noise makers 2.5metres away from the sensor laying flat on the ground, rotated 45 degrees at a time with each angle being an averaged for 60 seconds. 4.4, shows the orientation of the device, with the black arrow signifying the axis that WeMoS was spun around. A perfect result would show an absolutely flat line in the graph; however, it can be seen that the signal drops and rises as the device rotates. This change in RSSI is enough to move the calculated distance upwards of three metres.

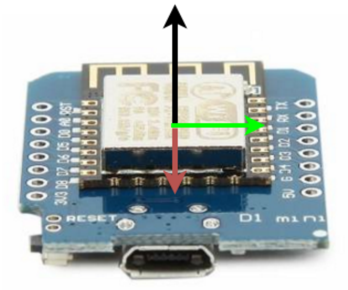


FIGURE 4.4: How the WeMoS was placed during the rotation test. The black arrow signifies the axis the device was rotated around.

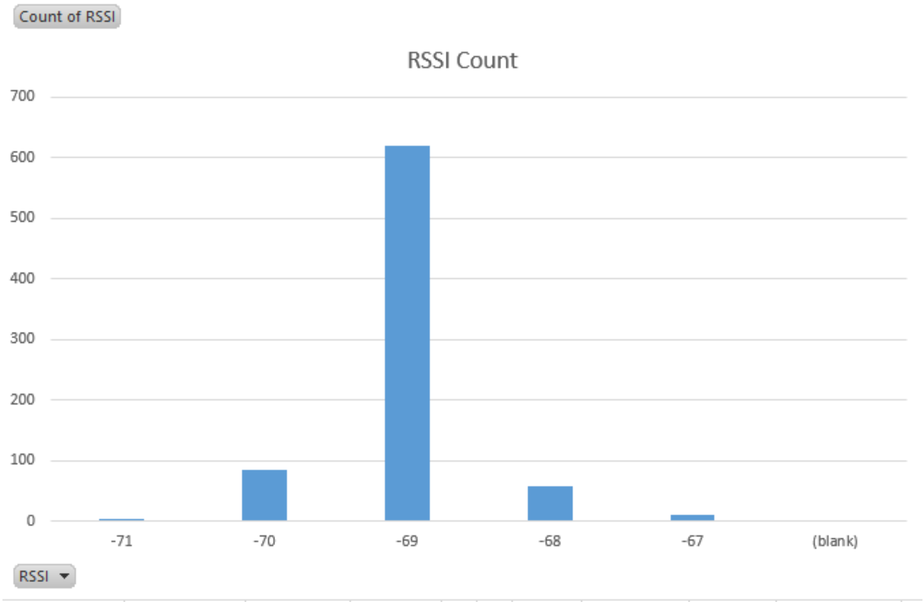


FIGURE 4.5: The occurrence of RSSI strengths. The best case scenario. 800 measurements of one second. 16 packets per a second averaged.

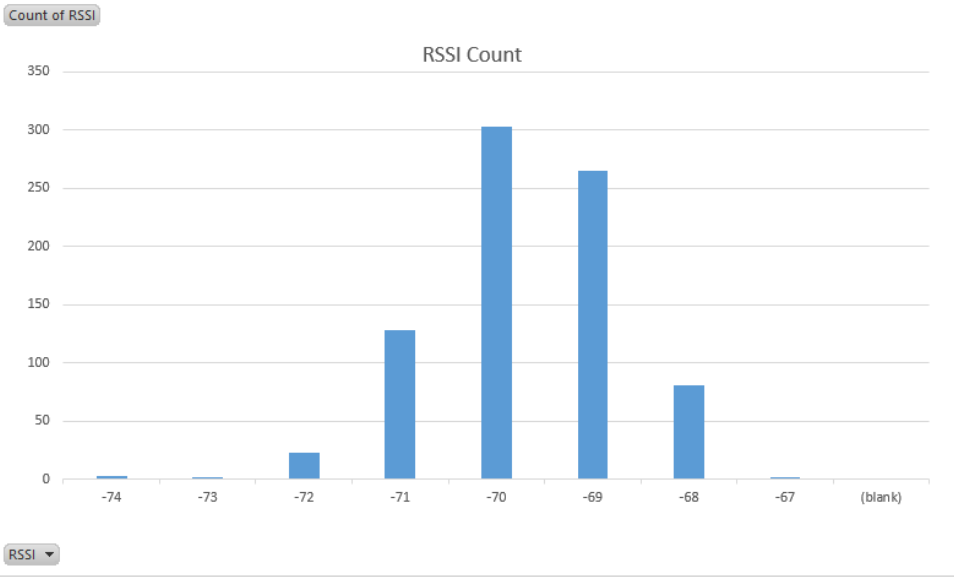


FIGURE 4.6: The occurrence of RSSI strengths. The worst case scenario. 800 measurements of one second. 16 packets per a second averaged.

The next test that was run was RSSI vs. time. The results from these can be seen in Figures 4.6 and 4.5. This test allows us to measure RSSI over an extended period and measure how often and how far it swings. The test was performed using a single receiver and multiple transmitters in a completely still room for 30 minutes. While the trial was repeated with many transmitters, almost all results were identical. Figure 4.5 shows the best case and Figure 4.6 shows the worst case scenario.

While there is a significant amount of noise in the RSSI measurement. Fortunately, it seems to be roughly normally distributed so a time-series average is likely to help. As a result, a sliding average is done over the last N samples and expiring samples that are greater than T seconds old. To cope with high data rates, the mathematics for this is performed locally on the sensors.

The test shows how unstable RSSI can be to measure. Even in a room with nothing moving, the best case scenario had the RSSI more than one dB off over 10% of the time. The worst case scenario shown in Figure 4.6 had an estimated RSSI value of -69.5dB; allowing for the signal to drop in the -70 or -69 basket, the RSSI value was still wrong almost 30% of the time. At this distance, a single digit above and below the correct value is a change of 40cm. These large errors are likely attributed to RF environment changes, such as a neighbour's network activity or microwave cooking and as such are completely unavoidable in the real world.

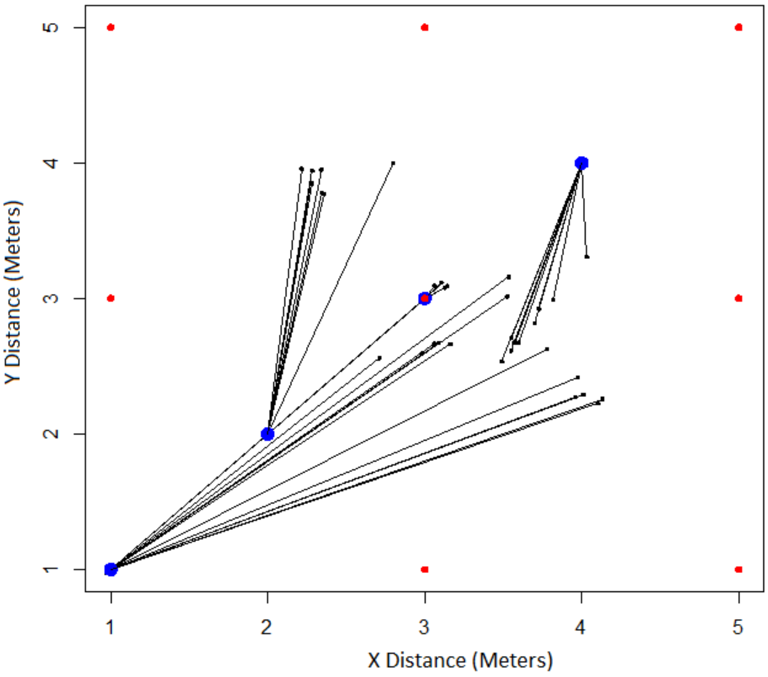


FIGURE 4.7: The jitter of a device's location over a period of time. Blue dot indicates the devices real location, black dots indicated calculated positions and red dots indicate sensors placed at the edges and center of the room.

Figure 4.7 gives a clear demonstration of how the centre of the room is more accurate than the edges. The blue dot in the centre of the room is almost dead accurate and the blue dots on either side at 2,2 and 4,4 have around 1.5ms of accuracy. However, the blue dot at 1,1 is calculated almost to be at the complete opposite side of the room.

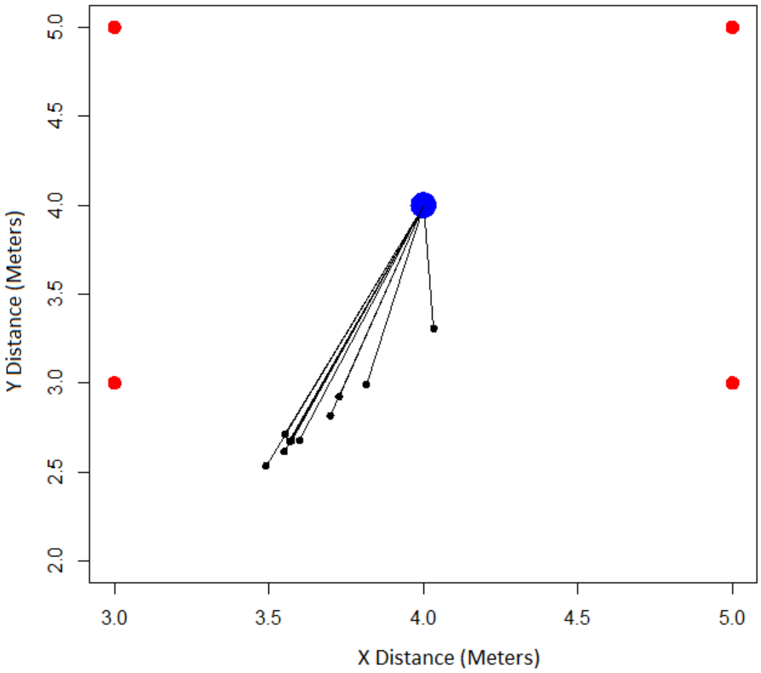


FIGURE 4.8: A zoomed in version of Figure 4.7. Showing only the data for the device located at 4,4.

4.3 Robust Regression

In an attempt to improve the accuracy of the system a Robust Regression approach was taken.

The idea is to run many tests, where each sensor, in turn, is excluded from the analysis. In the tests performed, trilateration was conducted independently on all N-1 sized subsets. Comparing the results to each other allows for the removing of inconsistent sensors and potentially improving accuracy. As if the results change substantially when a particular sensor is deleted, then that sensor is most likely an outlier.

From the results of the calculations, removing the sensor that caused the largest change in position from the standard algorithm caused a significant number of locations to improve with a few outliers. Overall these few outliers caused the average accuracy to decrease to around 1.98m.

Removing Sensor	Change In Position
1	1.5m
2	0.1m
3	0.2m
4	0.6m
5	0.3m
6	1.4m
7	0.3m
8	1.3m
9	0.2m

TABLE 4.1: Removing a selected sensor from the calculations causes its location to shift. Table shows the resulting distance change of that device.

Table 4.1 shows how far a device moves from the original calculations when removing a sensor. Note this movement may be in any direction, so the largest change is not always correct. As for which receiver should be removed, its seems like 60% of the time removing the largest change is the right way to go as you are removing an outlier. However, 40% of the time it appears to be either the second largest or third largest change. Table 4.1 shows an example output of removing a sensor, here removing sensors 1,6 or 8 caused a significant shift in the resulting position of the device. The data for N-1 sensors almost always looks the same but finding an algorithm that can accurately decide on what sensor should be removed is out of the scope of this thesis.

4.4 Calibrating the System

As each of the tests ran in the result were run separately, the designed algorithms were able to optimise each scenario to find the best possible values and constants for that test. If all the of tests are combined into one large set, and then optimised, the precise accuracy of the system can be calculated. Table 4.2, shows the results from this calibration, where All Sensors is using all nine sensors and N-1 Largest Change removes the receiver that caused the largest change in location.

Layout	Average Accuracy (Metres)
All Sensors	1.8
N-1 Largest Change	2.28

TABLE 4.2: The final accuracy values when all tests are combined

We can take this a step further for calibration. If we generate all combinations of scenarios, including every number of sensors between N and N-5, in virtually every possible combination of positions, then use R's optimise function to optimise all known locations of transmitters in all of those tests. The most optimal settings can be generated for almost all possible scenarios and layouts of the system. The resulting accuracy of virtually any setup will be around 2 metres, and the R optimised constants are below:

The actual RSSI to distance formula

$$RSSI = -10n \log_{10}(\frac{d}{d_0}) + A_0,$$

with the optimal constants: $n = 2.427$, $d_0 = 2.427$, $A_0 = -50.011$. It is interesting to note that n is 2.4 in these calculations while other sources as described in Section 2.1.1 expect a result around 22, perhaps the WeMoS scales its signal levels differently.
To perform the minimization maths for trilateration, a normalisation vector is used. The formula is below:

$$Error = \left| \sqrt{(Guess_X - Sensor_X)^2 + (Guess_Y - Sensor_Y)^2} - Sensor_{RSSI}) \right|^{4.04}$$

Adding an exponent to the end effects how the system handles outliers. The best way to describe this would be to say how much trust the system puts into each sensor e.g. If ten sensors line up, and one is far off, by default the algorithm will try to find the middle of them all. However, adding the exponent constant makes the algorithm heavily favour the location where they all intercept. The optimal value found while using R's optimise function was 4.04.

4.5 Testing the System

One of the major bonuses to using a very simple RSSI and Trilateration model is that is should require very little setup and calibration.

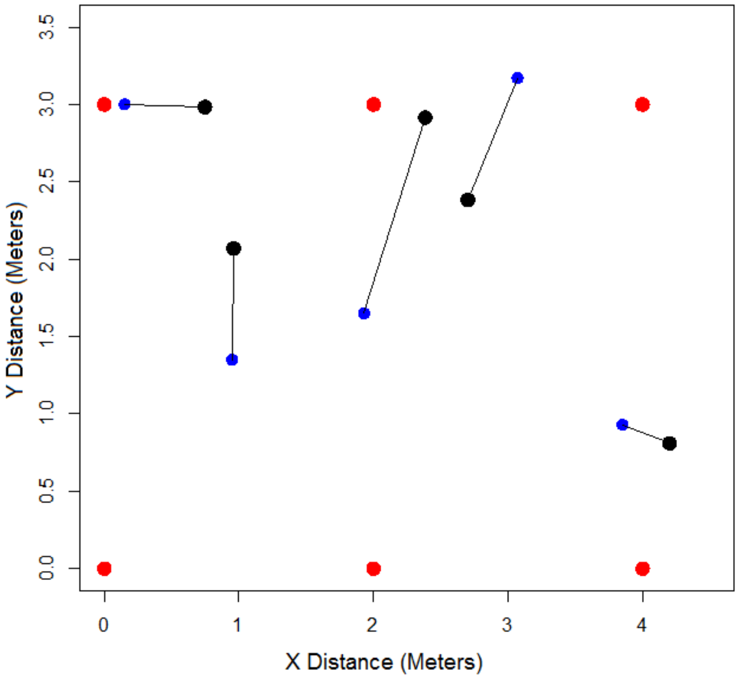


FIGURE 4.9: The results from the second round of testing. Blue dots are known locations of devices, black dots are calculated positions and red dots are sensors placed on the roof.

To test how well the system holds up, it was placed into a completely different environment, using the values calculated from the first round of testing. As can be seen in Figure 4.9, the results are very promising. The average accuracy of this setup was 0.78 metres. This is the best accuracy the system has ever had and is most likely because the second room had a higher density of sensors. The original room had each sensor covering 4M^2 of area; the new room has each sensor covering 2M^2. As seen in Figure 4.10, the second round of testing also has a lower amount of jitter.

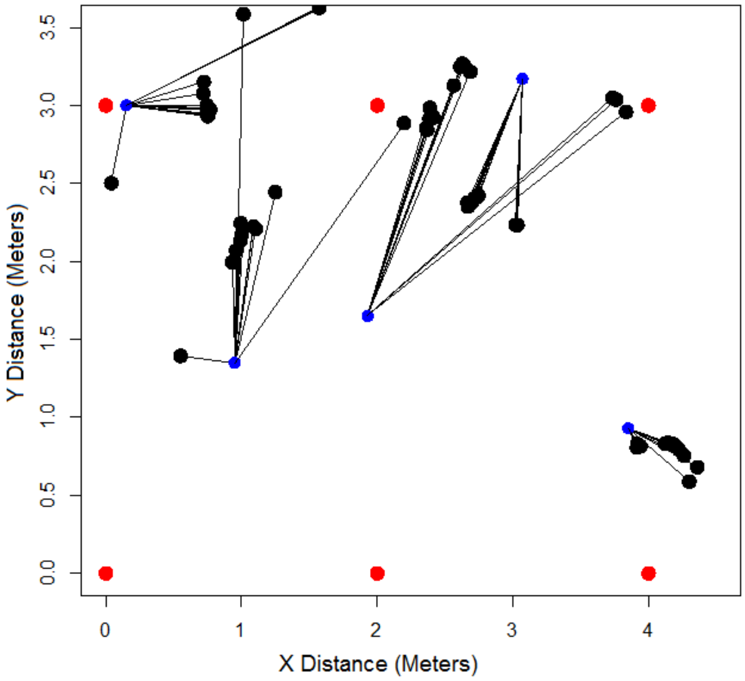


FIGURE 4.10: The jitter of devices from the second round of testing. Blue dots are known locations of devices, black dots are calculated positions and red dots are sensors placed on the roof.

4.6 Limited Budget vs Accuracy

The key benefit of this system is how well it can perform on such a small budget. Using the captured data, we can average every number of sensors and every possible location of the sensors to find the average accuracy, per number of sensors per M^2 of a room. Table 4.3 shows the average accuracy of both rooms.

Sensors per M^2	Average Accuracy (Metres)
0.500	0.850961
0.250	1.780643
0.222	1.927404
0.194	1.973552
0.167	2.056813
0.139	2.16142
0.111	2.286365

TABLE 4.3: The average accuracy of all possbile sensor locations

4.7 Surveillance

The system was able to produce an accuracy of within two metres with a sensor density of one sensor per 4M^2, and an accuracy of within one metre with a density of one sensor per 2M^2. However, the accuracy of this system is limited by multiple factors. The largest is multipath fading, where signals will bounce off walls and other objects causing signal strengths to vary wildly, pushing located distances of devices up to five metres away. Secondly, the non-isotropic radiation patterns of WiFi devices can cause shifts of a device's location by distances as large as 2.5 metres. Finally, background radiation such as noise from other wireless devices can push the location off by almost another a metre. Unfortunately, these are all completely unavoidable errors when it comes to RSSI-based localisation.

Still, this means the system is accurate enough to know what store a customer is in and even what isle they are in. Allowing stores to know exactly where people shop, how long they shop there, and what parts of the store they walk past. This information can be used to produce a full picture about a customer's likes and dislikes. Further, combining the data from multiple stores and locations can produce even more in depth and even creepy levels of insight into a customers lifestyle.

Based on the cost of an ESP8266 being around \$5 AUD. With a decent size shopping centre in Australia averaging around 100,000M^2. It will cost around \$250,000 for a complete shopping centre to have within metre accuracy indoor localisation. An accuracy of within two metres would cost around \$125,000. However, as most shopping centres only care about general location rather than metre accuracy, the whole shopping centre could be setup for as little as \$14,000. A small price to pay to be able to monitor every single customer's habits.

Indoor localisation, even when only accurate to within a room, creates a serious privacy issue. These systems allow for large companies to build full profiles for each of their customers. However, these systems can also be beneficial and life-saving when it comes to tracking Alzheimer's patients or small children. These systems are already pushing the limits of the law when it comes to consent and consistently fail to notify customers about the presence of such system. As these systems are completely undetectable, the only real solution is to educate people about these systems. Users must learn to disable the wireless radios on their devices or understand that they are most likely going to be tracked in almost all locations they visit.

Chapter 5

Conclusion

Indoor localisation is a rapidly developing technology. However, current technologies are prohibitively expensive, have inadequate levels of accuracy, and require too much calibration. As such a very simple system was proposed that used only basic RSSI measurements and trilateration.

Initially, this thesis ran three tests in a 36M^2 room using nine sensors. After calibrating the equations used to calculate location, an initial accuracy was found of around 2m. To determine the factors that affected the accuracy the most, a few tests were run to measure RSSI vs. Distance and RSSI vs. Time. The main culprit for inaccuracy is multipath fading, which caused inaccuracies of up to a few meters.

However, one of the largest benefits of a simple RSSI and trilateration system is that it does not require calibration. To test this, the system was setup in a smaller 12M^2 room with six sensors and only used the values calculated in the previous trials. The resulting accuracy was around 0.8 meters. An improvement most likely attributed to the higher density of sensors.

As these WiFi radio receivers have become small, inconspicuous, and easily concealed, these tracking and localisation systems are undetectable to those being tracked, and this presents a real privacy concern. The solution is relatively simple. In theory, as long as you have your WiFi turned off or have your phone set to flight mode (depending on the tracking technology used), the device will cease to send packets that can be tracked. However, the inconvenience of having to turn off your WiFi or enter flight mode every time you leave your house is enough to deter most people.

5.1 Future Work

A major limitation to the thesis was the number of sensors. The testing performed could have been improved by observing changes in accuracy caused by increasing the density of sensors. An additional avenue to test would of been to carry out the localization on a building-wide scale, to find the level of performance on the absolute lowest possible budget, as most establishments are not worried about meter accuracy but rather a system that is as cheap as possible but still capable of recognizing what area a customer spends their time in.

A path to research would be upgrading the ESP8266 to the ESP32 and running the localization via bluetooth to discover if there is any distinguishable difference in accuracy. Additionally, the ESP32 has two separate processors and can run calculations and WiFi connections simultaneously, potentially reducing the submit and lag time of the system.

There is a potential for the mapping and trilateration calculations to be performed in 3 dimensions. Giving data about the heights of shoppers and pedestrians, allowing for shops to find relations to how high certain products are placed on shelves. This could also give the system the ability to handle stairs and balconies.

It would be useful to design and build a small power efficient noise maker based off the esp8266. In particular to help keep track of small children or even animals in shopping centres or parks.

Indoor localisation is a fast growing technology and will be an interesting field to watch for many years to come.

Bibliography

[1] J. Xiong and K. Jamieson, “ArrayTrack: A Fine-Grained Indoor Location System,” *USENIX Symposium on Networked Systems Design and Implementation*, no. 279976, pp. 71–84, 2013. [Online]. Available: http://discovery.ucl.ac.uk/1329002/

[2] J. Yang and Y. Chen, “Indoor localization using improved rss-based lateration methods,” in *GLOBECOM - IEEE Global Telecommunications Conference*, 2009.

[3] P. Bahl and V. N. Padmanabhan, “RADAR: An In-building RF-based User Location and Tracking System,” in *Proc. IEEE INFOCOM 2000. The 19th annual conference on Computer Communications*, vol. 2, 2000, pp. 775–784.

[4] M. A. Youssef, A. Agrawala, and A. U. Shankar, “WLAN location determination via clustering and probability distributions,” in *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, 2003, pp. 143–150.

[5] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, “Indoor localization without the pain,” *16th Annual International Conference on Mobile Computing and Networking - (MobiCom ’10)*, p. 173, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1859995.1860016

[6] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, “Push the limit of WiFi based localization for smartphones,” *Proceedings of the 18th annual international conference on Mobile computing and networking (Mobicom ’12)*, p. 305, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2348543.2348581

[7] K. G. Coffman and a. M. Odlyzko, “Internet growth : Is there a “ Moore ’ s Law ” for data traffic ?” *Handbook of massive data sets*, pp. 47–93, 2002.

[8] K. Lawson, “The Hidden Dangers of Public WiFi,” no. October, 2014. [Online]. Available: http://www.privatewifi.com/wp-content/uploads/2015/01/PWF_whitepaper_v6.pdf

[9] H. Haas, “High-speed wireless networking using visible light,” *SPIE Newsroom*, 2013. [Online]. Available: http://www.see.ed.ac.uk/ hxh/Li-Fi_PAPERS/13_Spie_newsroom.pdf

[10] C. Pu, C.-h. Pu, and H. Lee, “Indoor location tracking using received signal strength indicator,” *Emerging communications for wireless sensor networks*, 2011. [Online]. Available: http://www.intechopen.com/source/pdfs/13525/InTech-Indoor_location_tracking_using_received_signal_strength_indicator.pdf

[11] Recommendation ITU-R P.1238-1, "Propagation Data And Prediction Methods for the Planning of Indoor Radiocommunication systems and Radio Local Area networks in the Frequency Range 900 MHz to 100 GHz," vol. 39, no. 3 Pt 2, pp. I–III, 1999. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/23463403

[12] Cisco, "Omni Antenna vs . Directional Antenna," p. 9, 2007. [Online]. Available: http://www.cisco.com/c/en/us/support/docs/wireless-mobility/wireless-lan-wlan/82068-omni-vs-direct.pdf

[13] DonDodge, "Indoor Location startups innovating Indoor Positioning," 2013. [Online]. Available: http://dondodge.typepad.com

[14] N. K. Sharma, "A weighted center of mass based trilateration approach for locating wireless devices in indoor environment," *Proc. Int. workshop Mob. Manage. Wirel. Access (MobiWac'06)*, pp. 112–115, 2006. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1164783.1164804

[15] A. W. Reza and T. K. Geok, "Investigation of indoor location sensing via RFID reader network utilizing grid covering algorithm," *Wireless Personal Communications*, vol. 49, no. 1, pp. 67–80, 2009.

[16] E. Navarro, B. Peuker, M. Quan, A. C. Clark, and J. Jipson, "Wi-Fi Localization Using RSSI Fingerprinting," *Test*, pp. 1–6, 2010. [Online]. Available: http://digitalcommons.calpoly.edu/cpesp/17

[17] Y. C. Y. Chen and H. Kobayashi, "Signal strength based indoor geolocation," *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333)*, vol. 1, no. 1, pp. 436–439, 2002.

[18] D. Lymberopoulos and J. Liu, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," *Proceedings of the 14th . . .* , no. Table 1, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=2737726

[19] H. Lim, L. C. Kung, J. C. Hou, and H. Luo, "Zero-configuration, robust indoor localization: Theory and experimentation," *Proceedings - IEEE INFOCOM*, pp. 1–13, 2006.

[20] J.-H. Lee, S.-J. Lee, Y. Park, K.-B. Yun, and K.-D. Kim, "RSS Based Indoor Localization Scheme Using GRNN and Virtual Grid-Points," vol. 1, no. 6, pp. 478–486, 2012.

[21] J. R. Jiang, C. M. Lin, F. Y. Lin, and S. T. Huang, "ALRD: AoA localization with RSSI differences of directional antennas for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 7–14, 2013.

[22] Y. He, A. Behnad, and X. Wang, "Accuracy Analysis of the Two-Reference-Node Angle-of-Arrival Localization System," *IEEE Wireless Communications Letters*, vol. 4, no. 3, pp. 329–332, 2015.

[23] S. Lanzisera, D. Zats, and K. S. J. Pister, "Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 837–845, 2011.

[24] M. Pourhomayoun, Z. Jin, and M. Fowler, "Spatial sparsity based indoor localization in wireless sensor network for assistive healthcare." *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2012, pp. 3696–9, 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/23366730

[25] C. R. Comsa, A. M. Haimovich, S. Schwartz, Y. Dobyns, and J. a. Dabin, "Source localization using time difference of arrival within a sparse representation framework," *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2872–2875, 2011.

[26] Innerspace, "The Unfulfilled Promise of iBeacons," 2015. [Online]. Available: http://www.innerspace.io/blog/2015/9/28/no-compromises

Appendix A

Risk Assessment

JAMES COOK UNIVERSITY
AUSTRALIA

Townsville Campus

College of Science, Technology and Engineering

Risk Assessment
Name of Test Indoor Localisation Thesis

Purpose: Thesis			
Operator:		**Duration:** March 2016 – October 2016	
SDS Attached: (Tick one)	Yes	[✓ No]	N/A
Major Hazard Types: (Tick at least one)			
Chemical		Mechanical	
Electrical		Thermai	
Environmental		[✓ Other:]	

SUMMARY OF RISKS

Specific Task/Activity	Potential Hazards/Consequences	Assessed Risk	Risk Control Measures	Reassessed Risk

Example:
Use of Soldering Iron
	Mild burns;
	Inhalation of vapour;
		LOW	Conduct in weil ventilated area;
			Avoid contact with skin;
			Do not breathe vapours;
			Wear appropriate PPE.
				LOW

Constant use of Computer	Repetitive Strain Injuries of the Wrists and Neck. Strain on eyes.	LOW	Take regular breaks. Use ergonomic keyboard and mice.	LOW
Handling of LiPo Batteries	Explosive/Flammable if charged incorrectly; Inhalation of vapour if explosion occurs;	MEDIUM	Use only a charger approved for lithium batteries. Never charge the batteries unattended. Charge your batteries in an open and ventilated area.	MEDIUM

SUMMARY OF REQUIREMENTS

Personal Protective Equipment	Safety Glasses
Is Training Required **If YES, please state requirements**	Yes/[✓ No]
Training Manual Location	

Revised: 4 May 2016

SUMMARY OF ACTIVITY

Using the Soldering Iron:

- Ensure the tip of the iron is clean.
- Wait until the iron has achieved operating temperature before using.
- After soldering is complete, allow the object the solder is applied to, to cool down before touching.
- Allow the soldering iron to cool down before storing.

Charging of LiPo Batteries:

- Use only a charger approved for lithium batteries
- Never charge the batteries unattended.
- Charge your batteries in an open and ventilated area.

ASSESSMENT:

OPERATOR (Student or Technician):

Jack Newman Jack Newm Date: 5/5/16 Contact No: ________

Name Signature

SUPERVISOR:

________ ________ Date: / / Contact No: ________

Name Signature

SAFETY ADVISOR:

John Renehan ________ Date: 5/5/16 Contact No: 14459

Name Signature

THIS FORM IS TO BE DISPLAYED IN THE IMMEDIATE VICINITY OF THE EXPERIMENT BEING UNDERTAKEN