Analyzing the performance of a WiFi tracking system

Alexandra-Ioana Alan

July 30, 2014

Acknowledgements

I would like to thank dr. Fabian Jansen, who had patience with me, answered my questions, and who helped me build the controlled experiments. I would like to thank dr. Martijn Gosselink and mr. Sandjai Bhulai, my KPMG and VU supervisors for their comments and feedback on the thesis. I would like to thank mr. Sander Klous for having faith in me and for giving me the opportunity to work with brilliant data scientists.

Next, I would like to thank Stojan Simonovski who saw potential in me at the Dutch Career Fair in May 2013 and who put me in contact with the amazing Alexandra Caraghiulea. Alex recommended me for this internship.

In the end, I dedicate this work to my mom, my dad, and Andrei.

Contents

Introduction

Davenport et al.[3] use as one of their premises the fact that data-based decisions help managers act in real-time and make better decisions which reduce bias, are cost-efficient, and easy to replicate. In addition to this, data can be modeled to predict future situations based on past events.

A challenge for a retailer/producer/wholesaler is to know when and how to adapt to the needs of clients. Thus, by using the knowledge extracted from datasets, managers can make informed decisions about the product placement, pricing, promotion, and profitability(Loraine Charlet et al.[4]).

Data are being collected and accumulated at a dramatic pace nowadays across a variety of fields(Fayyad et al.[1], Shaw et al.[2], Davenport et al.[3]). In addition to this, businesses try to develop intelligent ways of acquiring this customer data into large databases. According to Shaw et al.[2], useful marketing insights are sometimes hidden and undiscovered in these databases which can be translated into customer characteristics based on their purchase patterns. Using data mining algorithms and techniques, knowledge can be extracted and used for gaining competitive advantage. This knowledge can critically influence the marketing decisions and can improve business relationships.

Background and/or business context of the problem

One of the most compelling services of the KPMG platform represents Location Aware Services(LAS). LAS enables performing analysis on collected data and delivering business insights. This represents a novel method of gathering important customer data.

The business potential value of LAS will be studied on the Chep conference datasets, which are explained below and a dataset which will come from controlled experiments. These datasets contain WiFi data extracted through the mobile antenna signals. These data refer to the calculated location of people that were present at the conference at different time moments and locations.

The two main datasets were recorded at the Chep conference at the Beurs van Berlage (Amsterdam) in October 2013. The space of the conference was a large area where the people could walk and where they were not forced to choose certain paths for visiting, because the actual hall did not have a shape or walls. There were approximately 500 people visiting this conference and the recordings come from three different days.

The controlled experiments will be designed and implemented at the KPMG

headquarters and will be used for further understanding how the devices perform and whether some potential insight can be added to the one extracted from the main datasets.

Anticipated added value of the placement for the host organization/department

The motivation for the KPMG WiFi-efforts is to have a show-case that is able to demonstrate the following to the potential clients:

- Proving that tracking can technically be done
- Proving that this can be done while respecting privacy regulations
- Testing/calibrating the setup
- Developing analyses
- Showing the added value(the purpose of this research)
- Understanding the procedures w.r.t. privacy regulations a company needs to go through to get this working

The purpose and intended output (deliverables); success criteria

There are two algorithms developed by KPMG Big data & Analytics team for determining the location. The performance of these algorithms has not been tested before. One of the research questions that this thesis will try to answer is whether these algorithms are performing well and whether they provide accurate, reliable data given the conditions of the recording. In addition to this, we will try to explore whether there exist any business benefits for any potential clients and what are the potential use-cases of this system.

Thus, the main goals will be the following:

- Find according to the literature the most important user-requirements for a location-aware system
- Discover what are the most important parameters and conditions that need to be taken into consideration for the analysis
- Identify the challenges of building this system from both theoretical and practical perspectives
- Analyze the performance of the two developed algorithms and detect which one performs better and in which situation
- Design and implement controlled experiments that assess the performance of the drones
- Determine the limitations of the entire project
- Identify how this type of system can be used for potential clients and what are the potential implications of its use

Problem statement, including any formal preconditions

At this moment, there is no certitude of how well the system performs and how accurate the system is. This issue will be also tackled by analysing the recorded data and the stored logs during the conference.

It may be the case that the system would have to be configured for specific situations depending on the actual area of analysis. For example, what would be interesting to find out is whether the actual models for calculating the locations are performing well. Thus, controlled experiments will be designed and implemented in order to understand how good the devices perform. In the end, it is important to know whether the drones are properly configured and whether they record data in a timely manner, data which is accurate and can be used for extracting insight.

The problem, as stated before, represents a mixture between technical and business perspectives. From a technical perspective, we need to solve and implement the following elements:

- The accuracy of the recorded data of the customer/client/visitor/device

  Limitations:

  - The different capabilities of the mobile devices, such as: mobile antenna(signal strength), (supported) communication protocols, etc.
  - Lack of sufficient data of some recorded devices due to inactivity of the device in the chep data

- The meaning of the absence of data between the time of two consecutive recordings which represents one of the challenging issues, e.g.:

  - Dwelling time
  - Walking time
  - Missing data + dwelling time
  - Missing data + dwelling time+ walking time

  This can be interpreted as:

  - Is someone present, but not detected?
  - Is someone absent?

  Limitations:

  - Insufficient reference data

- For the data visualization, we will focus on two directions:

  - **Data visualization for a single MAC address** - important for measuring the accuracy of the recordings of the system and of the models, which will be performed on a specified time interval. In addition to this, we will create:

    * Animations of the path the MAC address follows

    - Generated histograms of x and y coordinates for detecting the resolution
- **Data visualization for multiple MAC addresses** - important for understanding the behavior of customer/clients/visitors on an aggregated level, which will be performed:
    - On a specified time interval(due to the large number of records)
    - On a specified day level
    - Generated histograms of the difference between the time of two recordings
- Comparison between the two developed algorithms based on data visualization module
- Focus on the analytics that can be derived, such as:
    - Number of detected, missing, arrived, and departed devices
    - The areas of interest and the average dwelling time for them
    - Typical behaviors of customers/clients/visitors/devices
- Design and implement controlled experiments for getting more insight
- Build a solution for describing the behavior of the customers/clients/visitors/devices within two consecutive detections(e.g. "presence probability").

Tools

The programming language that is used for implementing these analysis is Python with its libraries: Matplotlib, NumPy, sciPy, Pandas, Pymongo. The database in which the data is MongoDb, which is NoSQL database. It does not have a typical table format as SQL, but instead it has a BSON format(dynamic JSON documents).

Structure of the thesis

The thesis is structured in several chapters. Chapter 2 contains the comparison between the KPMG algorithms and assesses their performance. Chapter 3 proposes two models for calculating the probability of detecting mobile devices. Chapter 4 presents the experiments for extracting insight in how the mobile devices communicate with the routers in different situations. Chapter 5 refers to identification of the limitations and improvements of the Fitter algorithm. Chapter 6 includes the analysis of the system based on a framework from the literature. In the end, chapter 7 presents the limitations, conclusions, and the contributions to the literature.

Chapter 1

Literature review

1.1 Location aware systems

A smart-phone is an appropriate device to infer user context, because the data on the frequent interactions between users and their devices can be collected using various kinds of embedded sensors. For example, smart-phones can generate data through Internet connectivity which can be used for the location detection. This location information forms a core context in the pervasive computing environment[20].

According to Gu et al.[45], an indoor location aware system or indoor positioning system(IPS) considers only indoor environments such as inside a building. The author defines an IPS as a system that continuously and in real-time can determine the position of something or someone in a physical space such as in a hospital, a gymnasium, a school etc. An IPS should work all the time unless the user turns off the system, offer updated position information of the target, estimate positions within a maximum time delay, and cover the expected area the users require to use an IPS.

The position location of a smart-phone can be used for different scenarios. On one hand, one advantage could be public safety issues[19]. On the other hand, it can be used for social-context information. For example, a company scouting for locations to display its advertisements can obtain useful information on various places frequented by its customers in a certain time interval[8].

Several approaches are used to determine user location. According to Prasithsangaree et al.[19], these approaches are either by using a special infrastructure for positioning such as the global positioning system(GPS) or by enhancing communication infrastructures to determine the location of users.

GPS is, however, a common solution in open, outdoor environments. Prasithsangaree et al.([19]) state that GPS is not suitable for indoor areas, because of the lack of coverage. In addition to this, according to the same authors, it represents an expensive solution in terms of the costs of labor and capital for implementing a specialized infrastructure for detecting the

position indoor. In addition to this, previous research([18]) shows that a GPS signal is available only 4.5% percent of the time during a typical user's day. This suggests that average users spend much of their time indoors, where GPS service is normally restricted.

The WiFi positioning system is an effective alternative to GPS for indoor environments. An approach for detecting indoor position represents the wireless communication infrastructure to determine the location of users within the network. However, by comparing them with the outdoor, the indoor environments are more complex. In this case, there are various obstacles, for example, walls, equipment, human beings, influencing the propagation of electromagnetic waves, which lead to multi-path effects. Some interference and noise sources from other wired and wireless networks degrade the accuracy of positioning. The building geometry, the mobility of people and the atmospheric conditions result in multi-path and environmental effects(Gu et al.[45]).

According to Koyuncu and Yang[40], indoor positioning systems locate and track objects within the buildings and closed environments. These systems use wireless concepts, optical trackings or ultrasonic techniques. In addition to this, Gu et al.[45] identify other technology options for the design of the location aware systems such as infrared (IR), ultrasound, radio-frequency identification(RFID), Bluetooth, sensor networks, ultra-wideband (UWB), magnetic signals, vision analysis and audible sound.

Pahlavan et al.[44] describe how a wireless geolocation system should look like, which can be found in Figure 1.1. The main elements of this system, according to them, are a number of location sensing devices that measure metrics related to the relative position of a mobile terminal(MT) with respect to a known reference point (RP), a positioning algorithm that processes metrics reported by location sensing elements to estimate the location coordinates of MT, and a display system that illustrates the location of the MT to users. The location metrics may indicate the approximate arrival direction of the signal or the approximate distance between the MT and RP. The angle of arrival (AOA) is the common metric used in direction-based systems. The received signal strength(RSS), carrier signal phase of arrival (POA), and time of arrival(TOA) of the received signal are the metrics used for estimation of distance. As the measurements of metrics become less reliable, the complexity of the position algorithm increases. The display system can simply show the coordinates of the MT, or it may identify the relative location of the MT in the layout of an area. This display system could be software residing in a private PC or a mobile locating unit, locally accessible software in a local area network (LAN), or a universally accessible service on the Web. Obviously, as the horizon of accessibility of the information increases, design of the display system becomes more complex.

1.1.1 Positioning principle

Koyuncu and Yang[40] mention that the developed solutions for the indoor or outdoor position of objects are based on triangulation, trilateration,

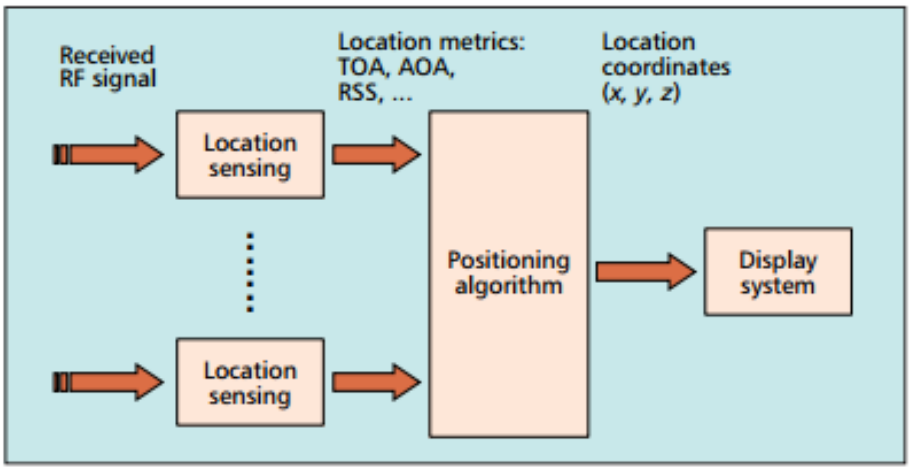


Figure 1.1: The diagram of a wireless geolocation system, according to Pahlavan et al.[44]

and multi-lateration methods using light[42], ultrasound, or radio signals, which provide positional information. However, Nuaimi and Kamel[12] argue that the main positioning techniques are: triangulation, scene analysis and proximity, and trilateration, which will be briefly described.

Triangulation is the process of determining the location of a point by measuring angles to it from known points at either end of a fixed baseline, rather than measuring distances to the point directly(trilateration). The point can then be fixed as the third point of a triangle with one known side and two known angles. According to Thomas and Ros[36], triangulation is a common operation not only in robot localization, but also in kinematics, aeronautics, crystallography, and computer graphics.

Scene analysis is another principle of positioning in which fingerprint is used(Nuaimi and Kamel[12]). The authors define a fingerprint as the unique characteristic or collection of characteristics of the scene, which works by collecting information from the scene and compare it with the existing database match for each scene.

Proximity principle is mainly used in Radio Frequency based systems, where a grid of antennas with fixed locations within the building are used. When a person carrying a mobile device is detected, the closest antenna represents the one taken into consideration when the objects' location is calculated. If more than one antenna detects the same mobile device, then the antenna that receives the strongest signal is used when determining the object's location(Nuaimi and Kamel[12]).

The main measuring technique used for the determination of coordinates in this thesis represents **trilateration**, which is a method to determine the position of an object based on simultaneous range measurements from three stations located at known sites(Thomas and Ros[36]).

In two-dimensional geometry, it is known that if a point lies on two circles, then the circle centers and the two radii provide sufficient information to

narrow the possible locations down to two. Additional information may narrow the possibilities down to one unique location[37].

In three-dimensional geometry, when it is known that a point lies on the surfaces of three spheres, then the centers of the three spheres along with their radii provide sufficient information to narrow the possible locations down to no more than two (unless the centers lie on a straight line). A method for determining the intersections of three sphere surfaces given the centers and radii of the three spheres is described and the entire derivation can be found in the appendix chapter based on the information explained in[37].

1.1.2 Applications

Sayed et al.[43] identified the main applications of location based systems. According to these authors, the forecasted business potential of these systems is tremendous world-wide as the number of users that own a cell-phone increases. Thus, they believe that the most important applications are the following:

- **Emergency services**, because a high percentage of calls to these services represent calls made by using a cell-phone.
- **Mobile advertising**, as mentioned before it can be used for tracking and attracting customers.
- **Asset tracking**, as it can be used for security services for locating a lost child, patient, pets, or assets.
- **Fleet management**, because it can help police forces identify cars, taxi companies, etc.
- **Location-based wireless access security** for avoiding the interception of digital information.
- **Location sensitive billing** which uses the location information of wireless users to offer variable-rate call plans or services based on the caller location.
- **Indoor navigation** for identifying places of interest

Pahlavan et al.[44] mention another important application in the public safety and military applications, where indoor geolocation systems may be important for tracking inmates in prisons, navigating policemen, fire fighters, and soldiers who need to complete their missions inside buildings. In addition to this, Gu et al.[45] write about various other scenarios such as fitness case and conference scenarios.

Zeimpekis et al.[13] affirm that the wireless tracking systems promise to enable the development of advanced mobile location systems in both the business-to-consumer(B2C) and business to business markets. The authors state that wireless positioning techniques have attracted much interest and research recently since they represent a core enabling technology for a continuously increasing number of mobile business applications.

1.2 Security and privacy requirements

The security of a system is the extent of protection against some unwanted occurrence such as the invasion of privacy, theft, and the corruption of information or physical damage. The quality or state of being protected from unauthorized access or uncontrolled losses or effects should be given to potential LAS clients. Safety is a property of a device or process which limits the risk of accident below some specified acceptable level. In addition, several aspects of privacy, such as approval by the user need to be considered[34].

The level of privacy influences the approval by the user:

1. How comfortable are users with their data (e.g., trajectory) being stored by another party?
2. Do users have legal concerns about their privacy?
3. If so, can private users be motivated to provide personal data?

Approval also includes the requirements for the system to allow certification by authorities, e.g., if there is a need for admissibility in court, the requirements for the system to deliver evidence should be given. Insurance companies should point out their policies concerning approval.

Kapadia et al. [23] study intensively the security challenges that arise from the location detection of the devices. The authors propose the following methodology with respect to the security challenges such as privacy, integrity, and availability, which should be individually tackled. In addition to this, Harris et al.[25] propose three basic privacy groups in which a participant can have one of the following profiles:

- *fundamentalist* - people that have **very high privacy concern**
- *pragmatism* - people that belong to a **middle group with balanced privacy attitudes**
- *unconcerned* - people that have **little to no concern about consumer privacy issues**

According to Mautz[34], the data privacy issue can be tackled from the perspective that users will want to control who may access to the information about themselves. Consolvo et al.[24] believe that the most important factors for participants to share their location are the following:

- *who* requests the information?
- *why* does this person request this information?
- *what* will be useful to this person?

Other important factors that influence whether participants want to disclose information of their location, which may not be important for this analysis, but play a key role in social media represent *activity & mood*. According to Consolvo et al.[24], participants were more willing to disclose

their location when they were “depressed” and they disclosed least when they were “angry”.

On one hand, in the same study, 56% of the participants were concerned with the privacy and security linkage of information from the location-enhanced applications for social relations. In addition to this, some of the participants were worried of a third party or unintended individual spying on their information or getting hold of their actual device.

On the other hand, Sadeh et al.[26] prove that people tend to be conservative about disclosures at first, but tend to relax their policies over time as they become more comfortable with mobile applications, in their case called Peoplefinder, and with how others are using it to find their location. In the end, the authors state that “this finding suggests that systems should help people stay in their comfort zones while also helping them evolve their policies over time”.

Even though the literature states that over time people stop being concerned of the privacy issues, the press does not support that. Forbes[28], one of the most important business newspapers, presented the story of a coffee shop which had to stop using the WiFi location aware systems, due to the privacy concern of its customers even though the tracking was transparent and there was also an opt-out feature built.

1.3 Projects in the literature

The literature is rich in articles that include projects related to location aware systems. However, these systems are small, pilot projects that record and analyze data for a small amount of time and these projects can be compared with the KPMG LAS system which is built to provide useful location-based insight for both users and clients.

Chon et al.[8] implemented a system called ”LifeMap” which represents a smart-phone-based context provider and a cost-efficient technology used for collecting indoor user context data. LifeMap uses inertial sensors in the smartphone to provide indoor location information. The information is combined with GPS and WiFi positioning systems to generate user context in daily life. The authors emphasize that there is need for such systems in order to improve the quality of services. The authors use as example finding the locations of a store most frequently visited.

Rekimoto et al.[5] developed a personal location-logging system called “LifeTag” that is based on the PlaceEngine location platform. The user of this system carries a small WiFi sensing device that periodically records surrounding WiFi fingerprint information (WiFi access point IDs and received signal strength). Later, this recorded information is converted into actual location logs by accessing the PlaceEngine’s WiFi location database.

Another interesting study was performed by Jiang et al.[7] who designed a remote pest monitoring system based on wireless communication technology. This system automatically reports environmental conditions and

traps pest in real-time. The acquired data was integrated into a database for census and further analysis.

Another interesting project is AnonySense, a privacy-aware architecture developed by Cornelius et al[22] for realizing pervasive applications based on collaborative, opportunistic sensing by personal mobile devices. Applications are allowed by AnonySense to submit sensing tasks that will be distributed across anonymous participating mobile devices. Later sensor data reports are received back from the field that are verified and anonymous. Their prototype is evaluated through experiments and two applications(a WiFi rogue access point and a lost-object finder).

1.4 Market research

There is an entire industry of location aware systems focused on WiFi tracking. Several companies have been identified as main players in the market. These companies will be briefly described based on the services they promise to offer, their target and main clients, and other interesting aspects that were available on the internet. The positioning techniques that the companies use are not disclosed.

A lead developer of such technologies represents Euclid Analytics[29]. Their office is located in San Francisco. According to their website, they translate anonymous device data into customer intelligence. Euclid looks at the visitor behavior based on WiFi data, shopping patterns of customer data, and calculates performance metrics. They target several industries, such as: executive, marketing, operations, IT. In addition to this, they offer calculations related to the most important KPIs: storefront conversion, average shop time, engagement rate, bounce rate, loyality, sales per day, conversion rate, sales transactions, average sales per week, outside opportunity, window performance, shopper engagement, store hours optimization, cross-shopping. On their website, they do not disclose any information related to their clients.

Another competitor represents PurpleWiFi, which targets its services to the following sectors: retail and leisure, hospitality, health and education, travel an transport, telecoms, marketing, public sector and commercial, and, lastly, but not the least, the event management. The company is located in Manchester, UK. According to its website[30], there are several business case studies where the company implemented the WiFi tracking and statistics: the Caesar Entertainment UK(casino industry), Orchards Shopping center(in West Sussex, UK), the Canterbury Cathedral(visitors tracking), Alhambra shopping center, Crystal Ski Holidays(implementation of WiFi tracking for ski resorts in France, Italy, and Austria), etc.

RetailNext is another in-store analytics company which combines the data from various sources such as: WiFi & Bluetooth devices, video cameras, point of sales data, staffing systems, weather, promotional calendars, payment cards and offers a web dashboard with custom reports combined with mobile apps based on which predictive analytics can be done. According

to their website[31], their customers are the following: Bloomingdales, Pepsico, Americal Apparel, P&G, and various other shops. The company started in 2007 with their headquarters in San Jose, California, but it also has a Dutch partner called WiFiProfs that offers the same services. The company has received last year many awards for being one of the most innovative companies.

Another Dutch group that offers a mixture between analytics and tracking technologies represents the Moreless group. Bluetrace is one of the companies that belong to the Moreless group umbrella that offers WiFi tracking solutions, which started in 2005. The company targets companies, public institutions, and governments and they offer a platform for customer loyalty and online marketing combined with tracking data. The main clients of Bluetrace are: Citroen, Bas Group, Febo, Seidensticker, Galleria Boromea, Ryanair, municipality of Haarlem etc. This company was in the news for not appropriately handling data privacy, they did not inform the visitors of the store that they were being tracked[56].

Polestar[27] represents a French company which offers indoor/outdoor positioning solutions with offices both in Toulouse, France and Palo Alto, US. The company was founded in 2002. Their technology is a mix between GPS, WiFi, Bluetooth Low Energy, and motion sensors that adopt to different environments and existing networks. In addition to this, they developed the NAO campus mobile application for improving the location based services. Their market solutions are split in four main categories: shopping malls& large retailers, transportation, museum and theme parks, and convention centers. For the large shopping malls, they offer both the mobile application for the customers, which can use the NAO Campus mobile app for finding out where their favorite shops are located, and the tools for extracting information about customer patterns, paths, and dwell time for the retailers. In the transportation area, their client is the Paris Charles de Gaulle Airport for which they used the same type of mobile application. They also work with two internationally renowned museums devoted to the promotion of science and technology and based in Paris, La Cité de la Sciences and La Palais de la Découverte.

Another European company that offers indoor location analytics is the German company Infsoft[32]. This is performed by combining multiple mobile sensors, such as: GSM, $3G/4G$(LTE), WiFi, magnetic fields, compass, air pressure, barometer, accelerometer, gyroscope, Bluetooth and GPS and they promise an accuracy of 1meter. In addition to this, the company offers interactive $2D$ and $3D$ maps.

Chapter 2

Comparison between the two developed algorithms

2.1 Description of the algorithms

Two algorithms were developed by KPMG for determining the location of devices based on WiFi radio signals. The two algorithms use different methods to determine the location of the packet source. In this chapter, the algorithms will be briefly described below and their performance will be compared from multiple perspectives: the resolution of calculated coordinates, the number of detected mobile devices, etc.

The algorithms are called the *Fitter* and the *Trilaturator*. Both take as input the same information: the source MAC address($sourceMAC$), the id string of the routers($drone\_id$), the raw WiFi signals that have been captured($signal\_strength$), and the time of the measurement($measurementTimestamp$). The dataset that contains all this information will denoted for simplicity the raw dataset. This information is recorded by the routers which communicate with the mobile devices and send the input information for the algorithms to the Kismet server. The entire process can be visualized in figure2.1. Then, the information is stored on the storm cluster using SSH. The algorithms run on the Storm cluster, which sends the output(e.g. sourceMAC, timestamp, and coordinates, Section 2.1.3) to MongoDB for storing.

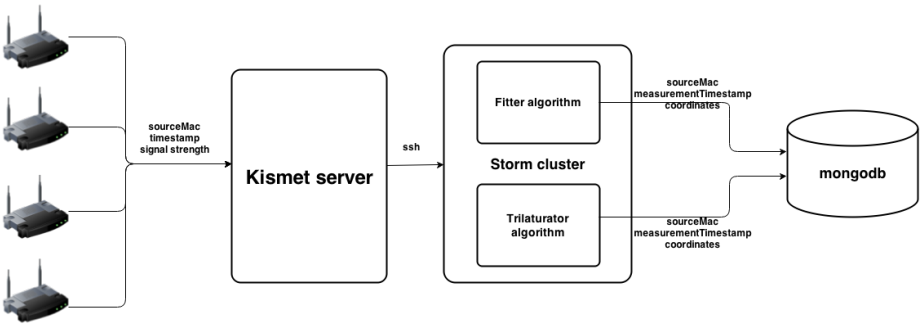


Figure 2.1: The algorithms

One of the assumptions the algorithms use is the existence of an inverse relationship between distance and signal strength. Hence, the higher the signal strength the shorter the distance and the lower the signal strength the farther the mobile device is from the routers.

2.1.1 The Fitter algorithm

The Fitter creates buffers of 10 seconds where it holds the data packets that come from different routers, then, these packets are sorted by time and by drone. Next, the algorithm tries to fit the values of the signal strengths of each 10 seconds bin using the weighted least squares technique due to the non-linear relationship between the signal and the distance. Then, by employing the Newton-Raphson method, an optimized curve is obtained for the given data. If the algorithm is able to fit the curve and to compute the coordinates, the status flag is $FITTED$, otherwise it is $UNKNOWN$ or $FAILED$, where the algorithm cannot provide coordinates. The reasons may be because it cannot converge after 100 iterations or that it does not have sufficient records or at least four different measurements from 4 different routers. More information can be found in the PhD thesis of Jan Amoraal[55].

2.1.2 The Trilaterator algorithm

In this algorithm, the raw data obtained from the drones is sorted by MAC address and by time. Then, for each MAC address, a buffer is created, which holds all the data records detected within a 10 second time frame. The next step consists of making combinations of 3 data packets each having different drone_ids. Then, the signal strength is required to have sufficient power(Signal_strength $\in$ [-90,0]). Thus, all packets with low signal strengths are dropped, which, for example, come from devices that are outside the area of interest.

Then, the signal strengths of the packets measured in dBms are translated into a distance estimate. By applying the trilateration technique, the x, y, and z coordinates are determined for each combination of 3 data packets.

If the number of combinations reaches 1.000(this threshold is introduced to reduce the computation time), the algorithm stops and computes the average values of the x, y, and z coordinates. Besides these values, the uncertainties of both x and y coordinates are also calculated and are equal to the standard deviation of all the computed combinations. These values will represent a record in the dataset that is being used for the analysis.

2.1.3 Dataset format of the algorithms

The datasets that we will do the analysis are the result of the trilateration process, which contain the following fields:

- **measurementTimestamp** - represents the date and time when the measurement was performed
- **sourceMac** - represents the hashed MAC address of a device
- **coordinates** - the calculated coordinates
  - **x** - represents the calculated x coordinate by the algorithm
  - **y** - represents the calculated y coordinate by the algorithm
- **error** - the error of the algorithms(the standard deviation of the algorithms)
  - $error_x$ - is the calculated error for the x coordinate of the algorithm(σ_x)
  - $error_y$ - is the calculated error for the y coordinate of the algorithm(σ_y)

In addition to this, a new variable was introduced to the dataset which is called $\Delta time$. The $\Delta time$ represents the difference between two consecutive timestamps t_{ij} of a hashed MAC addresses.

$$\Delta time_{ij} = t_{ij} - t_{i(j-1)}$$

,
where i corresponds to a certain MAC address and the j represents the index of the interval time of the detection. In addition to this, the first detection of a hashed MAC addresses is always initialized with 0. This variable is useful, because it can give us an indication of how long it takes between two consecutive detections of the same MAC address. The $\Delta time$ is calculated in minutes and the reason why this measurement unit was chosen was because it was easier to work with it.

Another two variables are introduced which are the **residuals** for the x and y coordinates. The residuals can only be calculated, if the actual coordinates are known, by using the following relationships:

$$residual_x = x - \tilde{x}$$
$$residual_y = y - \tilde{y}$$

where $\tilde{x}$ and $\tilde{y}$ represent the actual value of the coordinates, which can come from actual measurements or test cases, while the x and y represent the determined coordinates.

2.2 Comparison between the performance of the algorithms

The performance of the algorithms will be evaluated on the dataset from the CHEP conference by looking at the resolution for the x and y coordinates and the number of detected devices, which both depend on the device type as well as the time of the day. Therefore, an analysis will be performed by looking at how different types of devices behave, what is the resolution of the x and y coordinates, as well as the impact that the time of the day may have on the number of detected devices.

2.2.1 The resolution

The analysis of the resolution is based on the raw CHEP data. The logs contain an experiment where an iPad device was at a fixed position($x = 33.2m$ and $y = 18.6m$) for an hour and half. This experiment gives an indication of how often this device could ideally be detected if it was not moved and what the accuracy of the determined positions is compared to the known position. Moreover, we can discover whether the actual coordinates can be used for further analysis.

The resolution analysis is performed by comparing two different histograms: the residuals versus the errors of the algorithms for both x and y coordinates and the pull distributions for both coordinates.

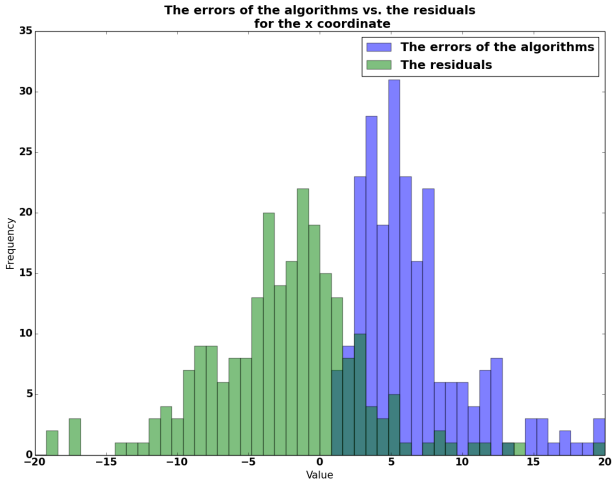
The residuals versus the errors of the algorithms

The histogram of the residuals versus the uncertainties of the algorithms of each of the coordinates gives information about how large the measurement errors are and whether there is a bias. The residuals were obtained by taking the difference between the coordinates calculated by the algorithms and the actual coordinates of the iPad device. In this sense, both algorithms will be compared by looking at their average x and y coordinates and the shapes of their residuals' versus errors' distributions.

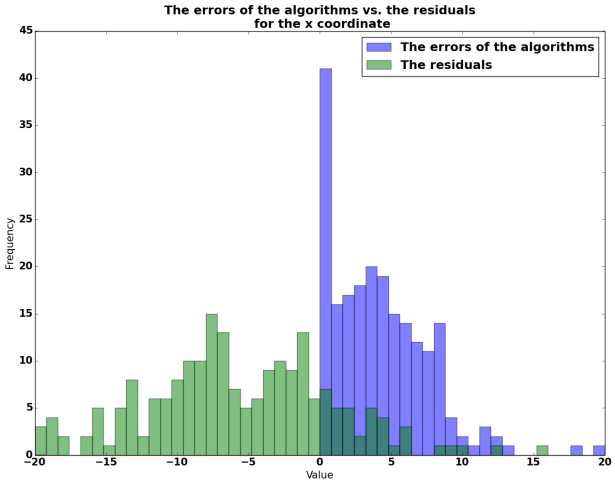
On one hand, as it can be seen in the case of the x and y coordinates of the Fitter algorithm(Figure 2.2a and Figure 2.3a), the distributions of the residuals follow a standard normal distribution slightly shifted to the left side and slightly shifted to the right side, respectively. The mean is not centered around zero, as expected. For the x coordinate, it has an approximate value of $-1m$, while for the y coordinate, it has an approximate value of $1m$. Thus, both means are biased. When looking at the uncertainties of the coordinates, it can be observed that the uncertainties on the coordinates are approximately $\pm 5m$ for both x and y.

On the other hand, the distribution of the x coordinate in the case of the Trilaterator algorithm does not follow the normal distribution(Figure 2.2b and Figure 2.3b). However, the distribution of the residuals for the y coordinate has the shape of a Gaussian distribution.

For further insight about the measurement errors, these plots should be interpreted together with the second plot of the Pull distribution, which gives much more information about the measurement errors.

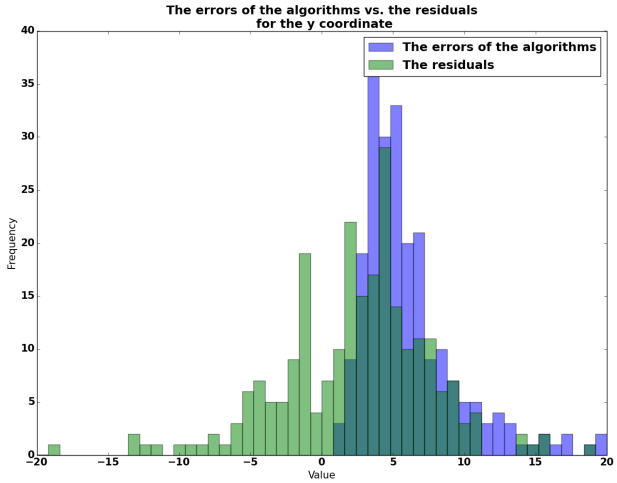


(a) Fitter algorithm

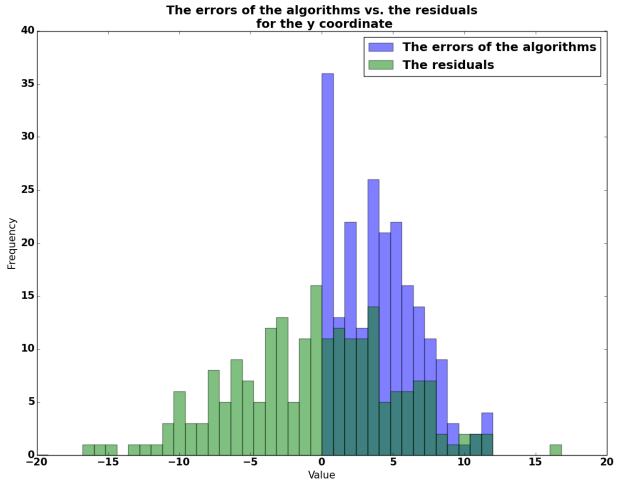


(b) Trilaterator algorithm

Figure 2.2: The residuals versus the error of both algorithms for the x coordinate



(a) Fitter algorithm



(b) Trilaterator algorithm

Figure 2.3: The residuals versus the error of both algorithms for the y coordinate

The Pull distribution of the x or y coordinate

Another method for analyzing the resolution of the calculated coordinates represents the pull distribution, which has the following formula:

$$pull_x(i) = \frac{(x_i - actual_x(i))}{\sigma_x(i)}$$

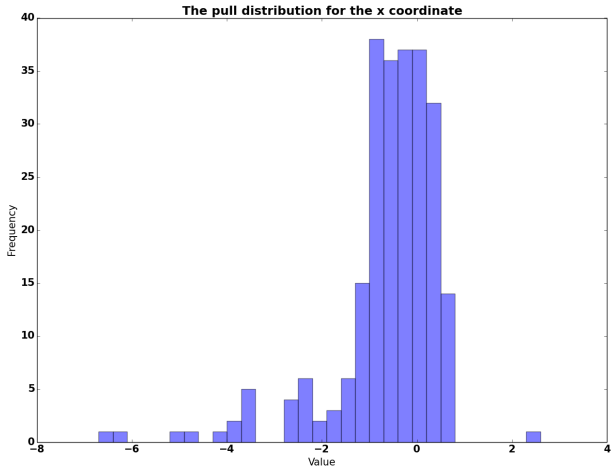
$$pull_y(i) = \frac{(y_i - actual_y(i))}{\sigma_y(i)}$$

where the x_i and y_i represent the determined coordinates, the $actual_x(i)$ and $actual_y(i)$ are the actual measurements of a MAC address, and the $\sigma_x(i)$ and $\sigma_y(i)$ are the errors of the algorithms.

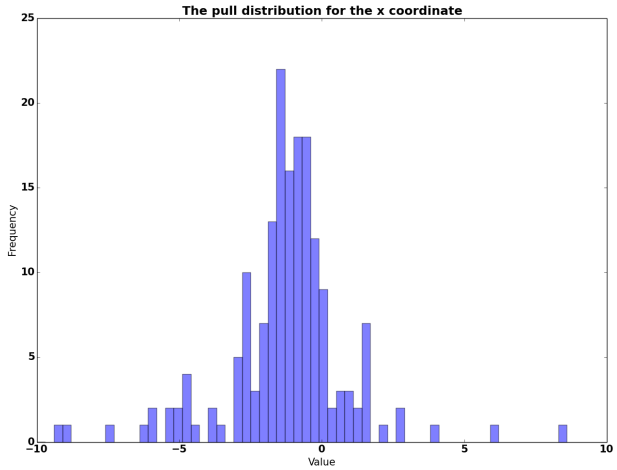
In Figures 2.4a and 2.4b and 2.5a and 2.5b, the Pull distributions of the both x and y coordinates are shown. The width of the Pull distributions gives more information about the measurement error. In the case of the Fitter algorithm, the center of the distribution for the x coordinate lies around $-1m$. Normally, this distribution should be centered around 0, however in this case it can be observed that the entire distribution is shifted to left which means that the mean is biased.

In the case of the y coordinate, the Fitter algorithms also has its mean not centered at $0m$, but shifted to the right side with a value of approximately $1m$. However, the Trilaterator algorithm has its mean centered around $0m$.

In the end, almost all three mentioned distributions resemble the standard normal distribution, except for the Trilaterator distribution of the x coordinate.

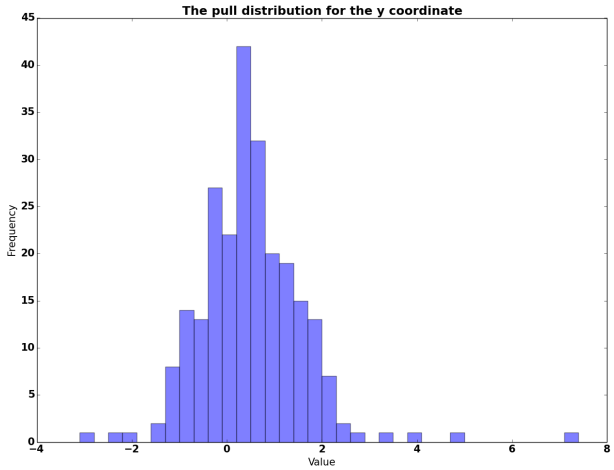


(a) Fitter algorithm

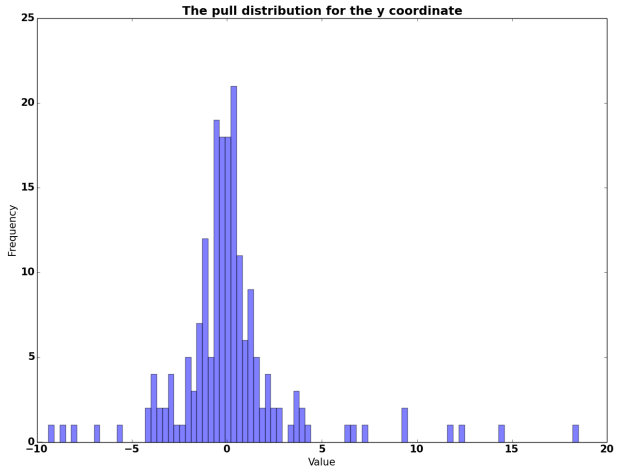


(b) Trilaterator algorithm

Figure 2.4: The Pull distribution of x coordinate of both algorithms



(a) Fitter algorithm



(b) Trilaterator algorithm

Figure 2.5: The Pull distribution of y coordinate of both algorithms

The animation of the reconstructed coordinates path in time

The animation of the reconstructed coordinates path in time is useful for looking at the path a device follows. This is performed from one detection to another. The plots from Figures 2.6a and 2.6b represent the positions of the detected coordinates of the iPad experiment, where this device was placed for one hour and a half without being moved. The labels of the points represent the values between two consecutive detections($\Delta time$). The initial value of the $\Delta time$ is 0 seconds.

It can be seen that the values of the Fitter algorithm are centered around

the actual point, which had the coordinates $x = 33.2m$ and $y = 18.6m$. However, it can be seen that there are also some points which deviate far from the values of the actual coordinates. A similar situation occurs also when using the Trilaterator algorithm. By comparing the performance of this algorithm with the one of the Trilaterator, it can seen that the points are better centered than the ones calculated by the latter, which has a larger spread around the centers of interest. This corresponds with the observations made with the residuals and Pull distributions. Thus, the Fitter algorithm behaves better than the Trilaterator algorithm.

Another interesting aspect may be the number of points that were recorded for the iPad devices, which can be seen in Table 2.1. The Fitter algorithm has detected more points for the iPad device than the Trilaterator algorithm which has 212 points, 32 fewer points than the first algorithm. In addition to this, the average value of the coordinate x is $30.7m$ in the case of the Fitter algorithm, which has a much higher value than the Trilaterator algorithm with $x = 26.8m$. However, the actual value of the x coordinate was $x = 33.2$. This means that for this coordinate the Fitter algorithm performed better than the Trilaterator, but still with some deviation from the actual value.

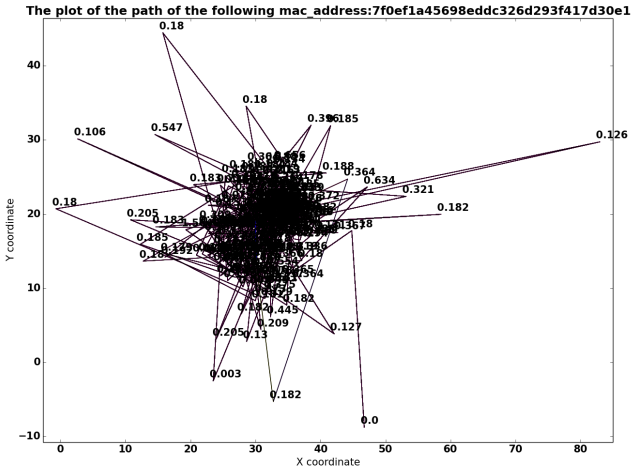
In addition to this, the same situation can be seen for the y coordinate for the $Trilaterator$ algorithm which has the average value of the y coordinate equal to $15.4m$, which is less than the $Fitter$ algorithm with an average value of the $y = 18.3m$ coordinate close to the actual value of the y coordinate $y = 18.6m$.

The root mean squared error will also be used for the comparison between the two algorithms, which constitutes a measure of the differences between values predicted by a model or an estimator and the values actually observed. According to Wikipedia[41], the RMSE represents a good measure of the accuracy, but which can used only to compare errors of different models for a particular variable.

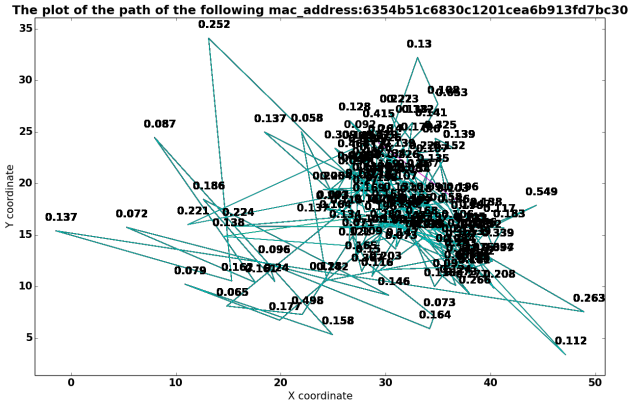
On one hand, the results for the x coordinate show that on average the deviation from the actual value is $7.8m$ for the Fitter algorithm, which is less than the RMSE obtained by using the Trilaterator algorithm, which has a RMSE of $9.8m$. On the other hand, in the case of the y coordinate the results show that the Trilaterator performs better than the Fitter with a RMSE of $5.7m$ compared to $6.4m$. As it can be seen, the calculated coordinates of both algorithms seem to be less and seem to have deviations from the actual coordinates. A possible explanation may be that the calculated coordinates are not reliable and that the models need improvement and much more testing. The large root mean squared error can be explained by the possibility that the actual coordinates were not very well recorded in the logs.

Element	Fitter algorithm	Trilaterator algorithm
The number of points	244	212
$Average_x(m)$	30.7	26.8
$Average_y(m)$	18.3	15.5
$RMSE_x(m)$	7.8	9.8
$RMSE_y(m)$	6.4	5.7

Table 2.1: Statistics



(a) Fitter algorithm



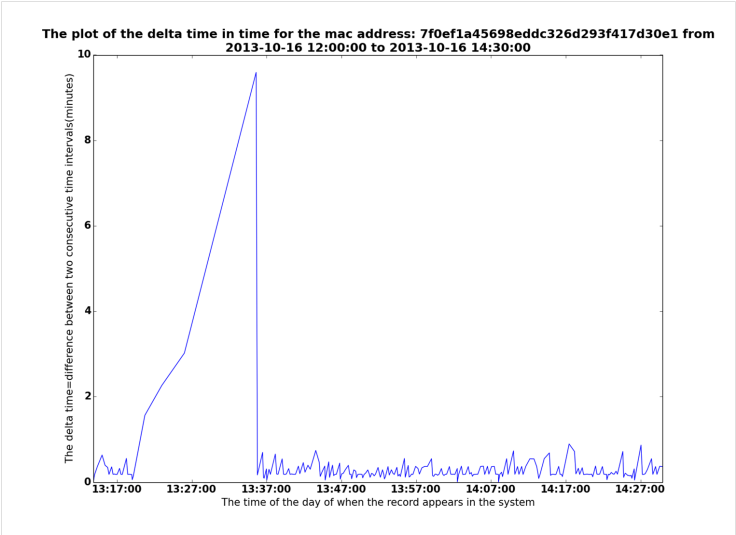
(b) Trilaterator algorithm

Figure 2.6: The animation of the detected coordinates of the iPad

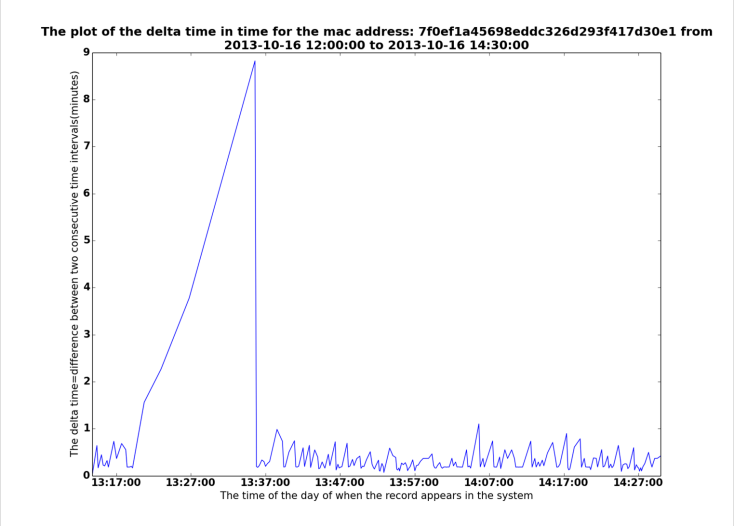
2.2.2 Device dependency

In this section, two different devices are analyzed with respect to two different quantities: the number of detections, but also how long it takes between two consecutive detections. These devices were recorded as tests during the CHEP conference. One of these devices was the iPad, while the other one represented a cell-phone which did a lot of streaming. The intention was to check whether the cell-phones behave differently.

By looking at plots of the iPad of both algorithms in Figures 2.8a and 2.8b, it can be observed that in most of the cases the device is being recorded quite often in time intervals less than one minute. However, in both pictures, it can be seen that there exists a large time gap between approximately 13 : 20 and 13 : 30, were the detections took more than two minutes with a maximum less than 10minutes. Because it is known that the iPad was not moved, it cannot be inferred that the device actually switched off. This situation can be either a technical problem or it may be due to an insufficient number of points based on which the trilateration technique can be applied and, thus, the algorithm could not provide a solution.

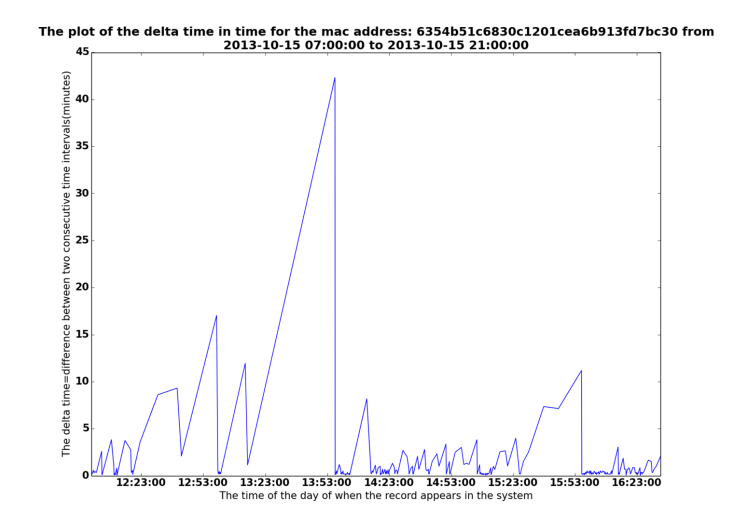


(a) Fitter algorithm

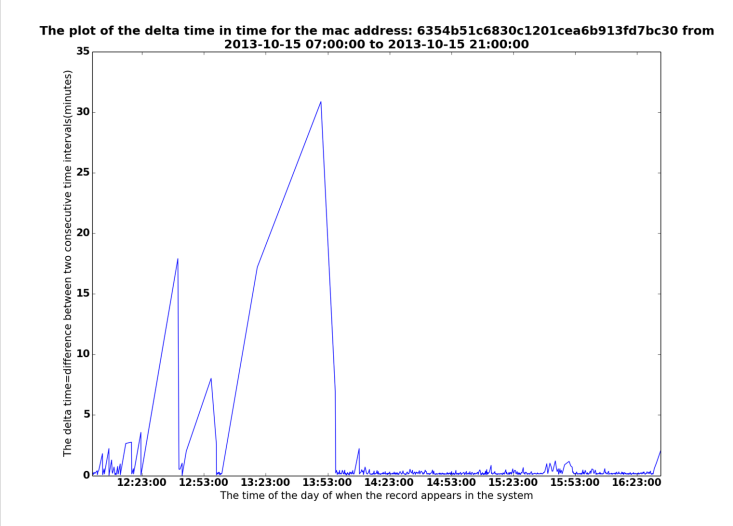


(b) Trilaterator algorithm

Figure 2.7: The delta time('time differences') and detections with respect to the moment in time and algorithm



(a) Fitter algorithm



(b) Trilaterator algorithm

Figure 2.8: The delta time('time differences') and detections with respect to the moment in time and algorithm

In the case of the second device, which represents a cell-phone with streaming data that was constantly moving, the time of the recording for this device was spread over an entire day and the detections can be observed starting from $07:00$ until $17:00$.

According to Figure 2.8b, the Trilaterator algorithm was able to calculate many more points with coordinates. For example, in total the Fitter algorithm has 1.098 records but only for 275 records the fitted coordinates were provided(the percentage of the fitted coordinates is approximately 25%). However, the Trilaterator algorithm was able to fit the coordinates

for 901 records. In the end, a positive aspect that can be drawn from both plots is that they have the same shape, even though the number of fitted points is very different from each other.

Another interesting aspect can be seen that starting from 13 : 00 o'clock, there is a very large gap of 45 minutes for the Fitter algorithm, while for the *Trilaterator* algorithm it was only of approximately 33 minutes. It can be the case that the Fitter algorithm did not have enough information to actually calculate the coordinates for that particular detection. Another potential explanation for this large time gap between detections may be that the device may have moved away from the area where the location aware system was installed which is in accordance with the logs of the company for this specific device.

Due to the absence of reference points, the resolution of these detections cannot be determined. Moreover, it is not possible to determine whether the actual information provided by the coordinates extracted from the algorithms is relevant and can be used. This issue is tackled Chapter 4 and Chapter 3, where controlled experiments are performed.

2.2.3 Number of devices in time

In this subsection, we will analyze how the algorithms perform on an aggregated level and how many devices are actually recorded during the time of the day if this behavior is spread around on a day level between 07 : 00 until 21 : 00. It provides information related to the number of devices that are recorded during this period within a time interval of 5 minutes. This means that each value contains the number of unique devices seen every 5 minutes.

First of all, the figures of plots 2.9a and 2.9b have consistent shapes. It seems that the Fitter algorithm has detected more devices than the Trilaterator algorithm. In this plot(Figure 2.9a), there was no cut made such that, for example, devices which are detected less than 3 times are avoided. This means that all the seen devices are taken into consideration.

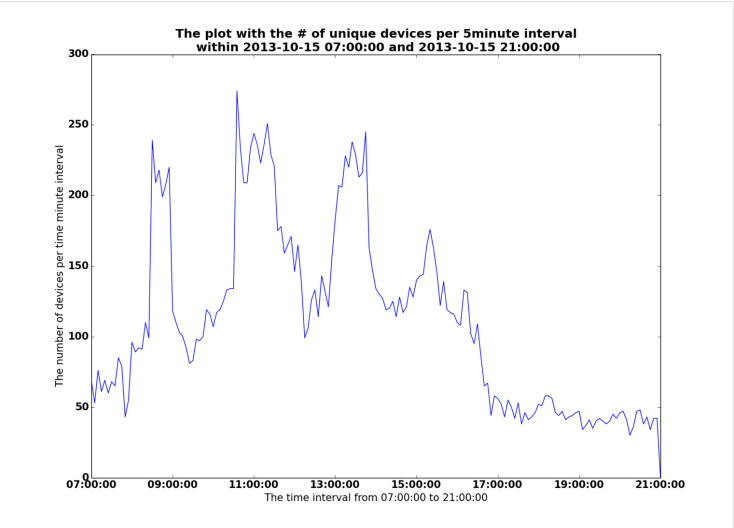
As it can be seen in the Figures 2.9a and 2.9b, there are four different peaks in the data, followed by dramatic decreases. These peaks and decreases last around one hour each and have almost the same number of detected devices.

Based on the schedule of the conference, the first peak corresponds to a dramatic increase which can be explained as the number of people arriving at the conference. The start time of the conference was 09 : 00. The decrease after the first peak corresponds to the fact that people stopped using their cell-phones or devices. Thus, this may mean that they attended the conferences and their devices became idle or they were turned off or they just left the area where the location systems were installed.

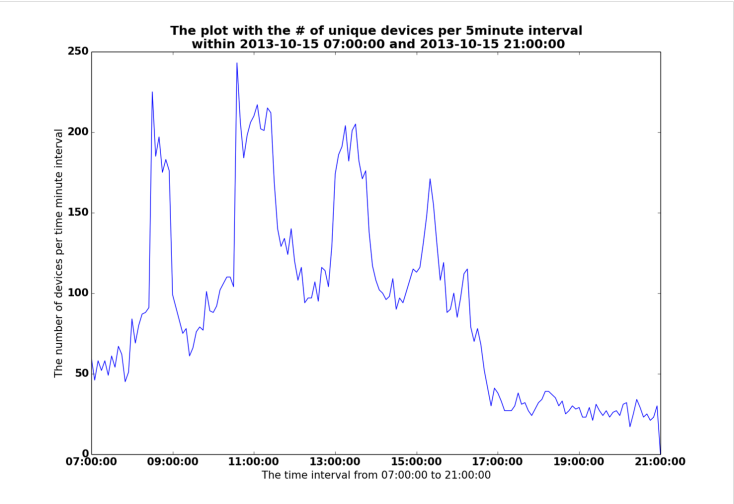
The second peak may be explained by the fact that after attending the conferences, people started checking their devices more often or streaming data. This increased activity produced thus more data. In addition to this, it may also indicate a coffee break taken after the first set of presentations.

The third peak corresponds to the lunch time, where people also come to the area which was recorded by the location aware systems. In addition to this, the third decrease may mean people attending presentations or people that are actually leaving the conference. An interesting aspect is that the number of detected devices during the first three peaks does not change significantly, which denotes that once people arrive their number does not actually change or if it does it is not significant.

The last peak does not have the same number of devices as the previous three. As mentioned before, there can be the situation that people left the conference and did not come back anymore. However, not all of them left yet. After this last peak, one can see a dramatic decrease of the number of detected devices which corresponds to the number of people leaving the conference for good.



(a) Fitter algorithm



(b) Trilaterator algorithm

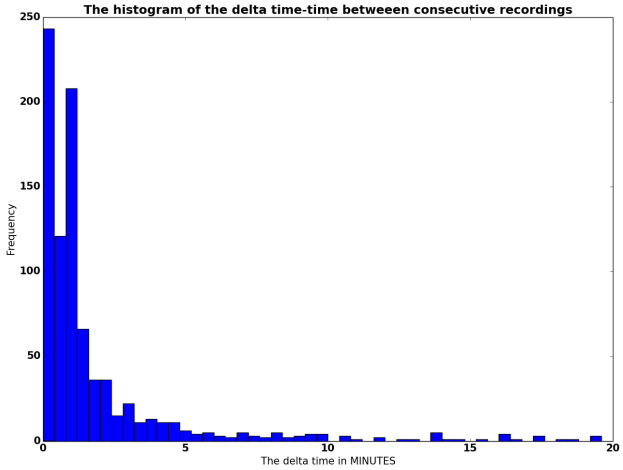
Figure 2.9: The number of detected devices during a day for both algorithms

2.2.4 Histograms of the time difference between detections

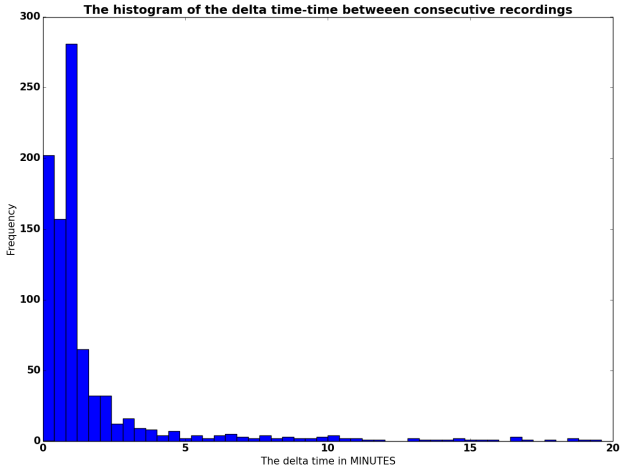
As previously mentioned, the $\Delta time$ represents a measurement for the time difference between two consecutive detections. In Figures 2.10a and 2.10b, the histograms of the $\Delta time$ can be visualized. In the case of both algorithms, the devices communicate in less than two minutes. An interesting difference that we can see between the two of them is that there seem to be more devices that have the delta time higher than 5 minutes in the case of the Fitter algorithm than for the Trilaterator algorithm. A potential explanation can be that the Trilaterator algorithm sometimes

calculates more data points than the Fitter algorithm, because the latter algorithm needs much more information. Hence, it has a smaller $\Delta time$.

The value of the $\Delta time = 5$ will be the threshold for the next chapter, where we will try to differentiate the devices that either are present($\Delta time <$ 5 minutes) or have left($\Delta time \geq 5$ minutes), even though they are present in the area with the mounted devices. This information will be also used for the controlled experiments, as it is important to understand in which situation we may draw the conclusion that a device is gone or missing.



(a) Fitter algorithm



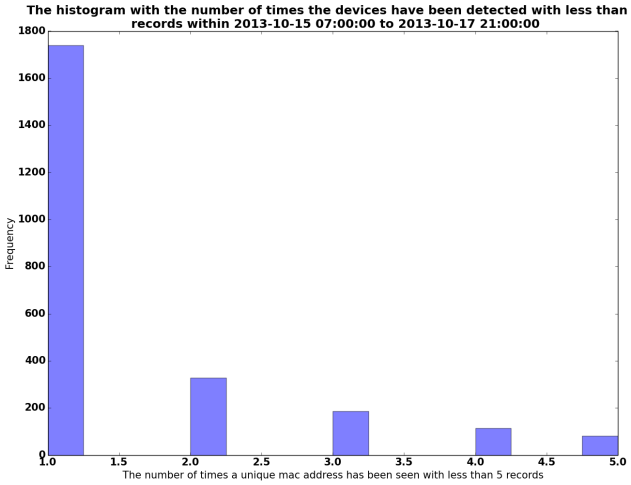
(b) Trilaterator algorithm

Figure 2.10: The histogram of the delta time between for both algorithms

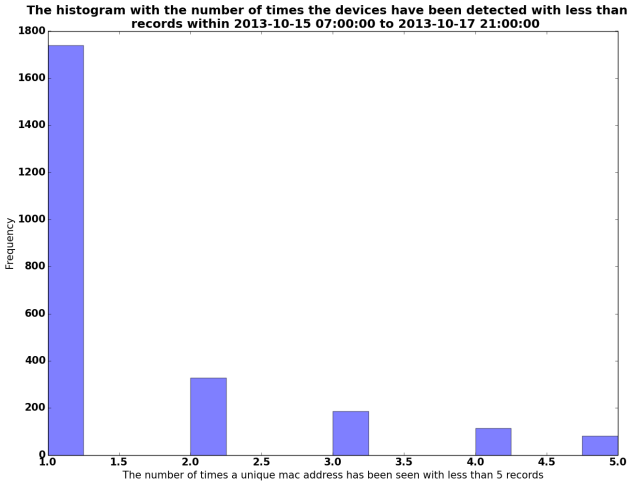
2.2.5 Number of detections per device

Ideally, a device can be detected every 10 seconds within the time interval between the moment it appears and disappears from the area where the location aware system is installed. However, this is not actually achieved due to the unknown state of the device which can either be idle or switched off, etc. In addition to this, the location aware systems record also devices which are close to the area under detection or devices of people walking outside. These devices usually have less than 5 detections with a very large $\Delta time$ between two consecutive detections.

What we would like to find is a potential threshold with respect to the number of records a device should have such that the dataset is cleaned of unwanted data. After several selected values of the threshold, the value which stood out was 5 records as it was the cut-off of the histogram with the number of detections per device. In order to increase visibility of the number of detections, we decided to split that histogram into two parts: less than 5 records and more than 5 records. In Figures 2.11a, 2.11b, 2.12a, and 2.12b, it can be observed that the histograms of the two algorithms look similar for both less and more than 5 records. This constitutes an interesting result as this seemed unexpected given the other discovered differences between the algorithms, such as: number of detected devices and the different number of detections per device(iPad animation). Moreover, the highest number of records for a device can be seen for MAC addresses which are detected only once. These MAC addresses should be eliminated.

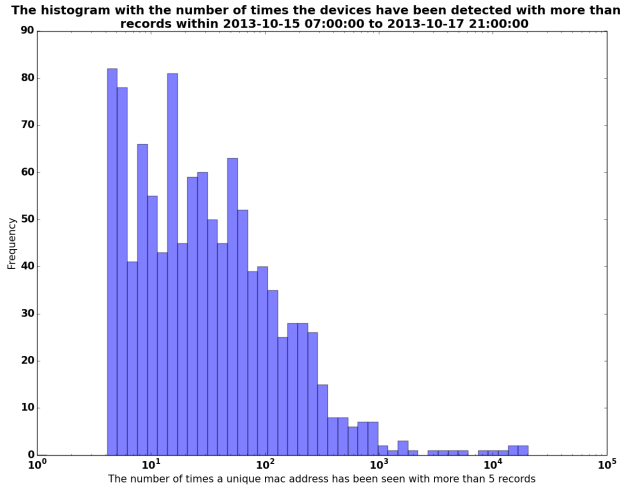


(a) Fitter algorithm

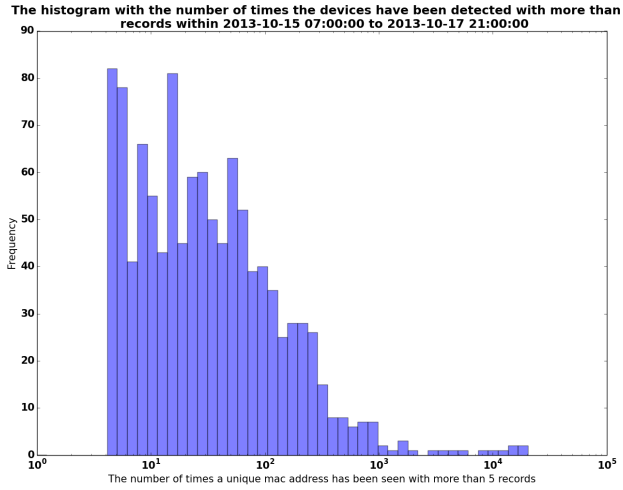


(b) Trilaterator algorithm

Figure 2.11: The histograms of the detected devices with less than 5 records for both algorithms



(a) Fitter algorithm



(b) Trilaterator algorithm

Figure 2.12: The histograms of the detected devices with more than 5 records for both algorithms

2.3 Conclusion

In this chapter, we performed the comparison between the *Fitter* and *Trilaterator* algorithms. The two have similar performance with respect to their resolution and efficiency. Nevertheless, the *Fitter* performs slightly better from a resolution perspective and, therefore, the rest of the analyses and models will be performed on the datasets obtained by applying the *Fitter* algorithm.

Chapter 3

Modeling the time dependency of detecting WiFi devices

In this chapter, we propose two models that calculate the probability of detecting a mobile device over time based on the Fitter dataset obtained from the CHEP conference. These two models are called the **Basic model** and the **Cut-off model**. The cut-off model uses the same principle of the basic model, but makes two additional assumptions:

- If somebody is not detected for more than 5 minutes, then that person left the area of analysis.
- A mobile device should have at least 5 detections.

These two thresholds of 5 minutes and records, respectively, were selected based on the results of Chapter 2 from the histogram of the delta time and the histograms of the detected devices.

The models also represent an attempt to identify in which state a device is during a time interval $\Delta time$. It is a challenge to determine in which state a device is, because the data recorded at the CHEP conference does not contain information on the state of the WiFi devices. The state of a mobile device may be: "active"(it communicates with the routers and it is in the area of the analysis), "gone"(left the area of analysis), "missing"(in the area of analysis, but not detected because of technical issues or the device is idle/switched off).

3.1 Basic model

Let there be a time period within the interval [$time_{start}$, $time_{stop}$] which can be either a day, half a day, or any other desired interval time. Both $time_{start}$ and $time_{stop}$ have the following structure: "yyyy-mm-dd HH:MM:SS",

where y stands for year, m for month, d is for day, H for hour, M for minute, S for second.

The *time_interval* represents the time interval for which we zoom-into the selected period ($[time_{start}, time_{stop}]$). The *time_interval* is expressed either on a second or minute level. In this section, we zoom-into the $[time_{start}, time_{stop}]$ on a 5 minute level, because we would like to get a first impression regarding the CHEP data.

Let an *Arrival* represent the first time a WiFi device MAC address is detected within a selected period of time. This means that before this detection there was no previous information related to this MAC address. A *Departure* is defined as the last time a MAC address is detected. Hence, one cannot find any later record in the database besides this one and we say that "it has left the system".

Figure 3.1 illustrates the times of arrival and departure within a time period $[time_{start}, time_{stop}]$. This pattern is similar for most WiFi devices. An exception constitutes the MAC addresses with a single record which have the time of arrival equal to the time of departure.

Figure 3.1: The diagram with the time of arrival, departure within a time period

A *"Missing"* MAC address represents a MAC address which has an *Arrival*, but it does not have a *Departure*. However, this MAC address is not being detected within certain time intervals. This situation can occur, due to several reasons: the drones do not function, the drones are busy, the device is not in the area of detection, or the device is idle.

Two arrays are defined with the size equal to the number of unique MAC addresses extracted from the dataset, *Arr* and *Dep*. For each MAC address $i \in [0, n-1]$, where n represents the number of unique MAC addresses, we compute the interval time between its arrival(Arr_i) and its departure(Dep_i) within the selected time period.

$$Arr_i = time\_arrival_i \in [t, t + time\_interval]$$
$$Dep_i = time\_departure_i \in [t, t + time\_interval]$$

where $t \in [time_{start}, time_{stop}]$ and *time_interval* represents the user selected time interval for which the analysis is made(e.g., 10 seconds, 5 minutes, etc.).

The next step is to introduce the number of arrivals and departures within a time interval denoted as $Arrivals_{[t,t+time\_interval]}$ and $Departures_{[t,t+time\_interval]}$. These two quantities can be determined by aggregating and counting the

MAC addresses an arrival or departure, respectively within that time interval, which is written as follows:

$$Arrivals_{[t,t+time\_interval]} = \sum Arr_i \in [t, t + time\_interval]$$

$$Departures_{[t,t+time\_interval]} = \sum Dep_i \in [t, t + time\_interval]$$

Besides the $Arrivals$ and $Departures$, the total number of unique detected devices within $[t, t + time\_interval]$ can be computed, which we denote as $Total\_detected_{[t,t+time\_interval]}$. The following relationship holds:

$$\begin{aligned} Total\_detected_{[t,t+time\_interval]} = Arrivals_{[t,t+time\_interval]} + Departures_{[t,t+time\_interval]} \\ + Active_{[t,t+time\_interval]} - In\_out_{[t,t+time\_interval]} \end{aligned}$$

where the $Active_{[t,t+time\_interval]}$ represent the MAC addresses that arrived in previous time intervals, but have not departed yet, and they are currently detected within the time interval of $[t, t + time\_interval]$. The $In\_out_{[t,t+time\_interval]}$ represents the number of unique MAC addresses that arrive and depart in the current time interval $[t, t + time\_interval]$. The reason why we need to correct with the $In\_out_{[t,t+time\_interval]}$ is that we add these MAC addresses twice: first for their arrival and second for their departure, but actually, they represent the same MAC addresses. Hence, they should be taken into consideration only once.

In Figure 3.2, the *Arrivals* and *Departures* are plotted together with the $Total\_detected$ for the CHEP data on which the Fitter algorithm was applied. The $time\_interval$ was selected to be 5 minutes and the time period is from $07:00$ until $21:00$ of the October 15^{th}2013, because we wanted to get a first impression on the CHEP data on a busy day. It can be observed that the highest number of arrivals are at the beginning of the day at $09:00$ o'clock, before the actual conference starts. Two other spikes can be seen at around $11:00$ and $13:00$. However, for the rest of the day, there seems to be a constant behavior of the arrivals.

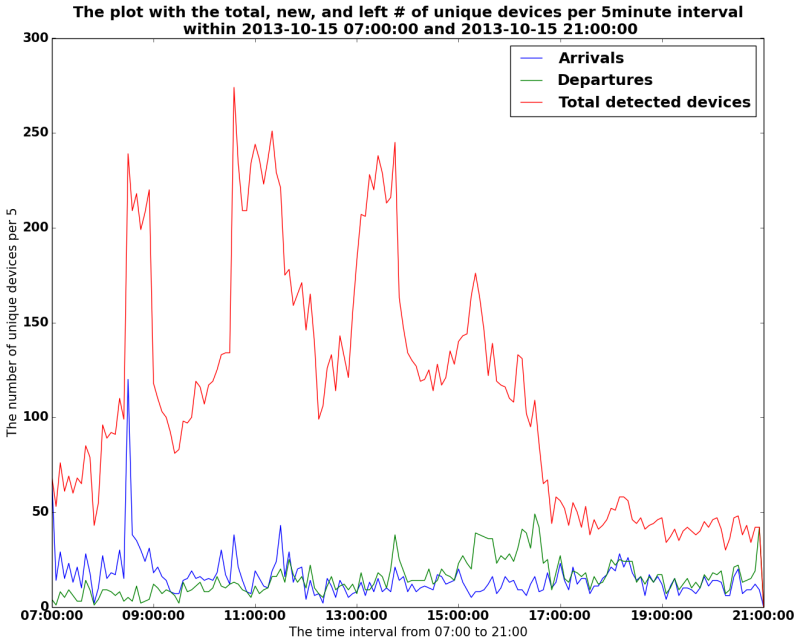


Figure 3.2: The total detections, arrivals, and departures for a 5 minute selected time interval from $07:00$ to $21:00$

The departures seem to have an opposite behavior, as expected. In the beginning, the number of departures is small, but different than zero. The reason is that other devices that pass by or access points are detected by the routers during the night. At the end of the day, some spikes for the departures can be observed which indicate that people leave the conference.

Next, we would like to determine the probability of detecting a device for a time period. Intuitively, the total number of devices($Total\_detected$) seems to be a good choice to be included in the calculation of this probability. However, it does not entirely reflect the correct number of MAC addresses that are present in the system for several reasons: the devices are in different states, the devices may not be seen by the drones due to technical problems. Thus, we introduce the ideal number of detected MAC addresses($Ideal$) that represents the number of detected devices at a certain time interval that should be detected by the system. Mathematically, the $Ideal$ can be computed as the difference between the number of arrivals minus the departures up to time interval $[t, t + time\_interval]$ as follows:

$$Ideal_{[t,t+time\_interval]} = \sum_{i=[time_{start},time_{start}+time\_interval]}^{[t,t+time\_interval]} (Arrivals_i - Departures_i)$$

The probability of detecting a device reflects the portion of detected MAC addresses which arrived in previous time intervals, but which have not left yet(*Active*) from the total number of MAC addresses which should have ideally been detected($Ideal$). The probability of detecting a device is computed as follows:

$$P_{detected\_devices[t,t+time\_interval]} = \frac{Active_{[t,t+time\_interval]}}{Ideal_{[t,t+time\_interval]}}$$

while $Active_{[t,t+time\_interval]}$ can be calculated based on the $Total\_detections$ formula:

$$\begin{aligned} Active_{[t,t+time\_interval]} = Total\_detected_{[t,t+time\_interval]} - Arrivals_{[t,t+time\_interval]} \\ - Departures_{[t,t+time\_interval]} + In\_out_{[t,t+time\_interval]} \end{aligned}$$

In the next plot(Figure 3.3), the ideal number of detected devices versus the *Active* is visualized. As it can be seen, the shown number of active MAC addresses has a similar shape as the $Total\_detected$: it has four spikes as in Figure 3.2. The $Ideal$ has a cumulative shape like a bell. It can be observed that the decrease of the $Ideal$ also corresponds to a decrease in the number of MAC addresses of the *Active*. This seems an interesting result, because it indicates that this decrease corresponds to people that leave the conference, which can be seen for both *Active* and $Ideal$ number of MAC addresses.

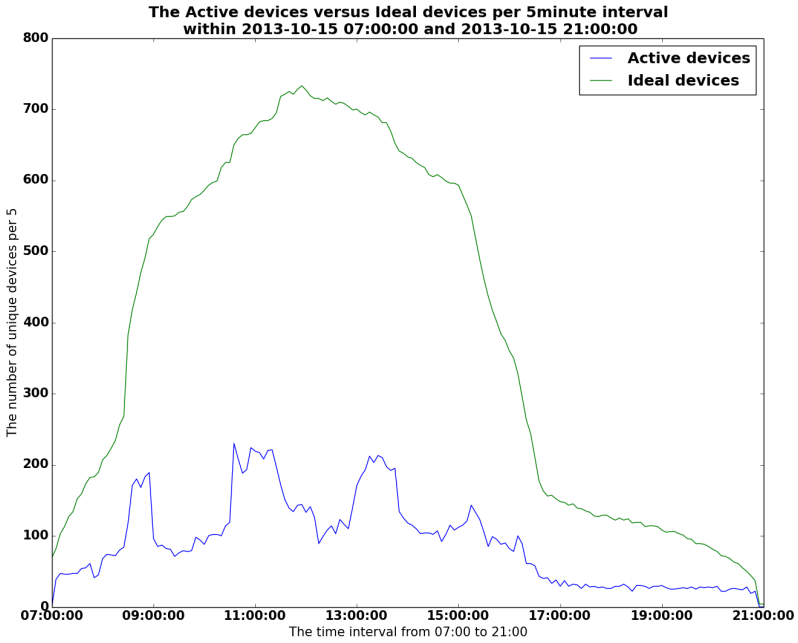


Figure 3.3: The ideal number versus the number of detected devices for a 5 minute selected time interval from $07:00$ to $21:00$

As mentioned before, the probability of detecting a device within a 5 minute time interval is calculated by dividing the *Active* by the $Ideal$. The result of this division is shown in Figure 3.4. As it can be seen, the shape of the probability is similar to that of the $Total\_detected$ and $Active\_macs$ with approximately four different peaks during the day that may indicate a time dependency. On average, this probability seems to be around 0.25 ± 0.09.

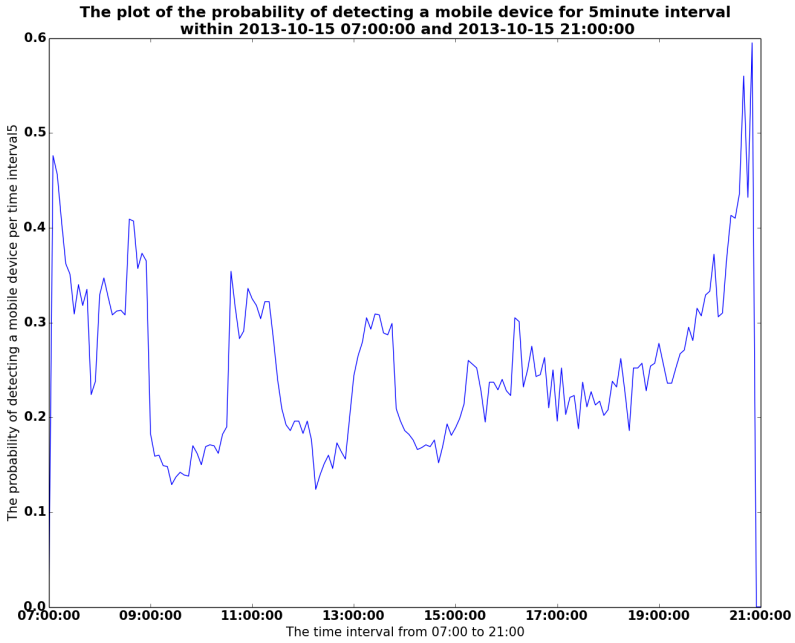


Figure 3.4: The probability of detecting a device within a 5 minute selected time interval from 07 : 00 to 21 : 00

In the end, the Basic model represents a first approach for computing the probability of detecting a device. Its advantages are that it is simple and intuitive. This Basic model uses a 5 minutes *time_interval* to compute all the quantities. This time interval is quite large. Hence, it is quite difficult to give any explanation regarding the devices that left the area of analysis and are expected to come back or that are still in there, but not detected. A solution to this challenge is to zoom in from a minute level to a second level, because we would like to identify how the Fitter algorithm computes coordinates for a MAC address every time interval. Besides this, the probability of detection may be underestimated, because the the *Ideal* is overestimated as no assumption was made regarding the fact that people may have left for some time the area under analysis. Thus, in the next section, we introduce a new algorithm that we expect to overcome these mentioned limitations.

3.2 Cut-off model

In this section, we present an alternative model to compute all the defined quantities from the Basic model. As discussed in the Chapter 2, the Fitter and Trilaturator algorithms use a 10 seconds buffer for computing the coordinates of a mobile device. Thus, the lowest *time_interval* in which

we can zoom into the dataset is 10 seconds. Besides this, we will perform the computations also on a 30 and 60 seconds level in order to evaluate the differences compared to 10 seconds.

A threshold on the number of missing intervals for a MAC address is introduced in this model, because the *Ideal* may be overestimated due to the fact that no assumption was made related to the people that may have left for a certain amount of time the area under analysis at the CHEP conference. Thus, we correct the *Ideal* and this corrected *Ideal* will be denoted as $Corr\_Ideal$. Moreover, this means that every time a MAC address is missing more than the threshold number of intervals then we assume that that particular MAC address is gone within that delta time and we correct this disappearance from the *Ideal* within that time interval.

$$Corr\_Ideal_{[t,t+time\_interval]} = Ideal_{[t,t+time\_interval]} - Correction_{[t,t+time\_interval]}$$

where $Correction$ represents a correction for the underestimation of the value of the detected devices. The corrected probability of detecting a mobile device is then:

$$P\_corrected_{[t,t+time\_interval]} = \frac{Active_{[t,t+time\_interval]}}{Corr\_ideal_{[t,t+time\_interval]}}$$

The *behavior of a MAC address* represents a compressed way to describe how a MAC address has been or not detected, which contains information about the time period expressed as the number of consecutive detected intervals followed by the number of consecutive missing intervals between its arrival and departure. This compressed way of describing a MAC address is efficient when calculating the statistics for very small time intervals as 10, 30, or 60 seconds. The reason is that within a day there are large numbers of 10 seconds time intervals and unique MAC addresses. It may be inefficient to go through every time interval for each MAC address. Thus, using the behavior overcomes this challenge.

For example, a behavior of a device can be observed in Figure 3.5, where we can see that the interval between [*Arrival*, *Departure*] covers 18 *time_intervals*. Its *Arrival* is within the 7th interval(the notation of the interval starts from 0) and its *Departure* is within the 24th time interval of *time_interval* size.

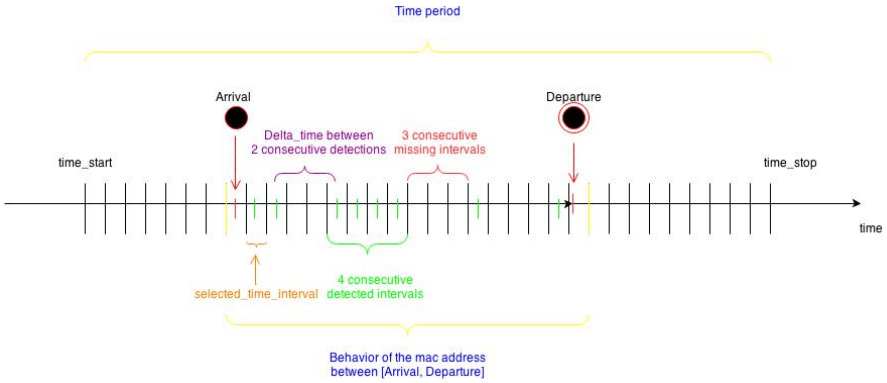


Figure 3.5: The behavior of a MAC address within a time period between $time_{start}$ and $time_{stop}$ split into multiple $time\_intervals$

Hence, we define the behavior of this MAC address as the collection of calculated consecutive and missing intervals between the time of arrival until the time of departure. We remind the reader that the number consecutive missing intervals($'m'$) constitutes the number of consecutive intervals with no detections of the MAC address under analysis, while the number of consecutive detection intervals($'d'$) suggests the number of consecutive intervals that have at least one detection within every interval. Hence, for the device in Figure 3.5, its behavior will be:

$Behavior = \{'d' :3,' m' :2,' d' :4,' m' :3,' d' :1,' m' :3,' d' :2\}$.

By construction, this collection always ends with a detection(that of the departure). In addition to this, we retain the interval numbers of the arrival and the departure(7th and 24th).

The algorithm behind the cut-off model has two parts:

1. The computation of the behavior of all MAC addresses based on the dataset of the Fitter algorithm
2. The calculation of the main quantities:
   (a) The total number of detected devices, the arrivals, and departures for every $time\_interval$
   (b) The active MAC addresses and the corrected ideal
   (c) The corrected probability of detecting a mobile device for every $time\_interval$

The first part deals with computing the behavior of all unique MAC addresses that can be found in the dataset. Thus, for each MAC address $i \in [0, n-1]$ in the dataset, which has more than 5 detections, the algorithm starts from its arrival and computes alternatively the number of consecutive time intervals with detections($consecutive$)and the number of consecutive intervals with missing detections($missing$) until the departure.

For each detection j of a MAC address i, if the time of that detection is within the time interval $[t, t + time\_interval]$, then the number of detections within $[t, t + time\_interval]$ is calculated. Otherwise, if the time of a detection is not within $[t, t + time\_interval]$, then the number of time intervals with missing detections are calculated, denoted as *missing*. The *missing* is equal to the difference between the time of the current detection and the lower bound of the current time interval t divided by the *time_interval* size. In this case, there are two situations:

- If the *missing* number of time intervals is greater than 1, then it means that the next detection is located at *missing* time intervals distance from $[t, t + time\_interval]$. If *consecutive* is greater than zero, then this number of consecutive intervals with detections is appended into the list of behaviors and the *consecutive* is initialized with 1. After the *consecutive*, the *missing* is also appended to the list of behaviors. The current time interval t becomes $t + missing \times time\_interval$.
- Otherwise, if *missing* is equal to 1, then the next detection is located within the next time interval. Thus, if the number of detections within this time interval is greater than 0, then the number of consecutive time intervals is incremented. The current time interval t becomes $t + time\_interval$.

As it can be seen, the algorithm goes only through the time intervals with detections of MAC address i, which increases the performance of the algorithm, because the intervals with no detections of that particular MAC are avoided. In addition to this, using lists as main data structures makes the computations faster and easier to deal with. The entire algorithm can be visualized in Section 1 parts 1 and 2. Besides the matrix of behaviors, the function also returns the list of the location of each analyzed MAC addresses, because their location in the dataset is not clearly determined and there is no simpler way of actually determining it.

The second part of the algorithm deals with calculating the corrections of the *Ideal* by considering "gone" the MAC addresses that have the Δ_{time} between two detections greater the *threshold*. These corrections of the *Ideal* need to be made for the entire time interval in which the MAC address is missing, at the exact position where the MAC address is not present anymore until it actually comes back. In order to calculate these corrections in an efficient way, the following elements are needed:

1. List of all behaviors, which contains the behaviors of all unique MAC addresses in the dataset.
2. List of arrivals, which contains for each MAC address i the number of time intervals after which its arrival occurs. Its size is equal to the total number of unique MAC addresses n.
3. List of departures, which contains for each MAC address i the number of time intervals after which its departure occurs. Its size is equal to the total number of unique MAC addresses n.

Algorithm 1: The computation of the behavior of all MAC addresses-Part 1

```
[1] Data: dataset['mac','time']
Input: time_interval, threshold_records, time_start, time_stop
list_missing=[];
list_detected=[];
list_all=[];
list_macs=[];
list_arrivals=[];
list_departures=[];
Calculate index= the array with first positions in the dataset of a
mac(Arr);
Calculate the n unique number mac addresses;
for i in n-1 do
    if Arr[i+1]- Arr[i] ≥threshold_recordsthen
        list_macs.append(dataset['mac'][i]);
        t=round(Arr[i]);
        list_arrivals.append((Arr[i]-time_start).seconds/(time_interval));
        list_departure.append((Dep[i]-time_start).seconds/(time_interval));
        missing=0;
        already=0;
        consecutive=1;
        c=0;
        list_all=[];
        for j in range(index[i], index[i+1]) do
            if t ≤ raw_data['time'][j] ≤ (t + time_interval)then
                c+=1;
            myalg
```

4. intervals - represents the total number of time intervals of user selected size between the $time_{start}$ and $time_{stop}$ and the formula is:

$$intervals = \frac{(time_{start} - time_{stop})}{(time\_interval)}$$

The difference between $time_{start}$ and $time_{stop}$ is expressed in seconds in Python. If we want to determine the number of intervals expressed in minutes, we divide by $60 \times time\_interval$.

The algorithm performs the following steps. For each MAC address i, the arrival and departure are extracted from their corresponding list. A variable *pos* retains the starting position of the consecutive intervals with detections, which initially is equal to the arrival of that particular MAC address. In addition to this, the length of the list with the behavior of MAC address i is calculated. Then the iteration over the list of the behavior of the MAC address i begins. A new variable k records the position of a detection as long as we have not reached the position of a missing interval and the $Total\_detected$ is updated. Next, if we have not reached the last position in the list of behavior for a particular MAC

Algorithm 2: The computation of the behavior of all MAC addresses-Part 2

```
[1] myalg else if then
    missing=(dataset['time'][j]-t).seconds/interval;
    if missing>1 then
        if consecutive>0 then
            list_all.append(('d',consecutive));
            consecutive=1;
        end
        list_all.append(('m', missing-1));
        t=t+missing × time_interval;
        c=1;
    else if missing==1 then
        if c>0 then
            consecutive+=1;
        end
        t=t+time_interval;
        c=1;
end
if consecutive>0 then
    list_all.append(('d', consecutive));
end
list_all_states.append(list_all);
end
end
return list_all_states, list_mac, list_arrivals, list_departures, list_index
```

address(last $'d'$ with the last consecutive intervals with detections), the variable *pos* is updated with value of the starting interval of the missing plus the number of missing intervals. The variable *pos* refers now to the next number of consecutive intervals(next d in the list of behavior of a MAC). Then, we check whether that particular number of missing intervals is greater than the *threshold* given by the analyst. The *threshold* is used, because we make an assumption that if someone actually left the area where the location aware system is installed for more than the *threshold* time intervals, then he or she is gone. Thus, for that particular number of missing intervals, the $Ideal$ needs to be corrected, and this correction is made by extracting this person/device from the counting. Next, the $count\_cutoff$ is updated for every time interval in the range of missing and we jump only to the intervals with consecutive detections. Otherwise, in case we reached the last element of the list of behavior for that particular MAC address, we do not correct anymore the position, because the last element does not contain any other number of consecutive missing intervals($'m'$). The entire algorithm can be visualized part 3.

Algorithm 3: The calculation the total detections and correction

```
[1] Input: list_all, list_arrivals, list_departures, max_distance, threshold
n=size(list_all);
Total_detected=array(max_distance);
count_cutoff=array(max_distance);
for i in range(n) do
    arr=list_arrivals[i];
    dep=list_departures[i];
    pos=arr;
    m=len(list_all[i]);
    j=0;
    while (j<m) do
        k=pos;
        while k ≤ (pos+list_all[i][j][1]-1) do
            Total_detected[k]=Total_detected[k]+1;
            k+=1;
        end
        if j < m-1 then
            pos=k+list_all[i][j+1][1];
            if list_all[i][j+1][1] ≥ threshold then
                t=k;
                while t ≤ (list_all[i][j+1][1]+k-1) do
                    count_cutoff[t]+=1;
                    t=t+1;
                end
                j=j+2;
            end
        else if j==m-1 then
            pos=k;
            j=j+1;
    end
end
return Total_detected, count_cutoff
```

3.2.1 Results

In this section, the results of the cut-off model for three different user *time_intervals* of 10, 30, and 60 seconds and their probability of detection are presented.

As we mentioned in the description of the cut-off algorithm, certain parameters need to be selected for this algorithm such as the threshold of records for a certain MAC address, the threshold of missing intervals, and the period of time. These decisions were made made based on the results from Chapter 2, where the comparison between the two algorithm was discussed.

Firstly, the *threshold_records* was set to 5. According to Figures 2.11b and 2.12a, it can be observed that there is a dramatic decrease of the number of devices with more records than 5. The idea behind selecting this threshold is that the dataset needs to be cleaned from MAC addresses which do not have sufficient detections and may damage the analysis and the model. In addition to this, this quality cut-off removed all the *In_out* MAC addresses which had the arrival and departure within the same time interval. Secondly, based on the results of the histogram of the delta in Figure 2.10a, the decision of selecting the *threshold* of missing intervals to 5 minutes was made. In the case of 10, 30, and 60 seconds *time_interval*, the threshold will be 30, 10, and 5 missing intervals, which all represent 5 minutes and which were obtained by dividing $\frac{5\times 60s}{time\_interval}$. Lastly, the period of time was chosen to be October $15th2013$ from $07:00$ to $21:00$. This period was chosen as a regular day of the conference, which can be split in $5,040$ intervals of 10 seconds intervals, $1,680$ intervals of 30 seconds intervals, and 840 intervals of 60 seconds intervals, respectively.

The results of the cut-off model can be visualized in Figures 3.6a, 3.7a, and 3.8a. As it can be observed, the plots show the ideal, the corrected, the total detections, and the active. The probability of detection is based on the corrected probability of detection.

Statistics	10 seconds	30 seconds	60 seconds
Average probability	0.337	0.542	0.713
Standard deviation	0.131	0.122	0.109
Number intervals	$5,040$	$1,680$	840
Threshold missing	30	10	5

Table 3.1: Statistics

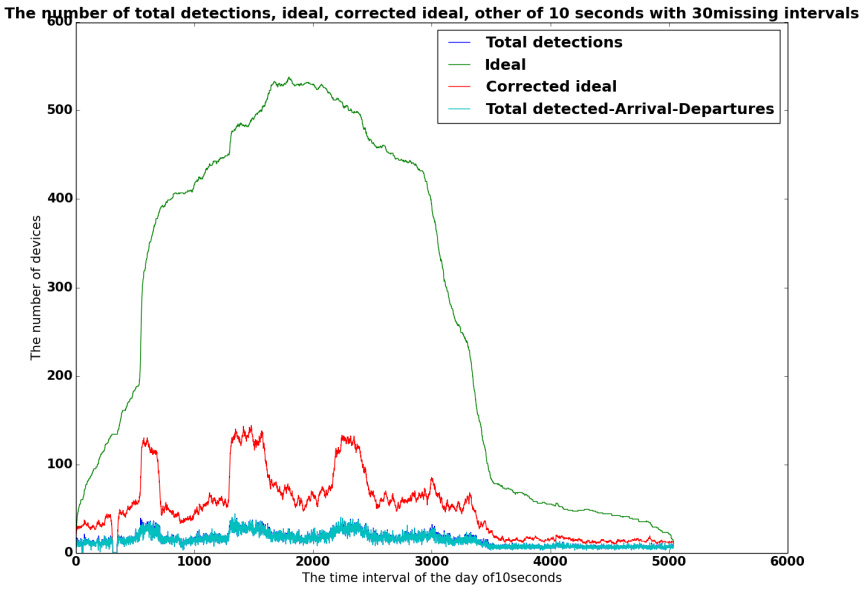
As it can be seen in Table 3.1, the average probability increases as we increase the size of the *time_interval*. This was expected, because the probability of having detections in a larger interval is higher than having detections in a smaller interval. In addition to this, it can be observed that the probability of detecting someone has a flatter shape now than in the basic model(Figure 3.4). However, the variations can still be seen with respect to the time period. Moreover, the original form of four major peaks is still present. This structure with four different peaks becomes less

evident if a higher time interval is chosen, for example 30 in Figure 3.7b. In the case of the 60 seconds time interval, the shape of the probability becomes flatter and the structure of the variations is less evident(Figure 3.8b). The reason between this change is that with a higher selected time interval, the results are more aggregated and fewer computations are performed with more data.

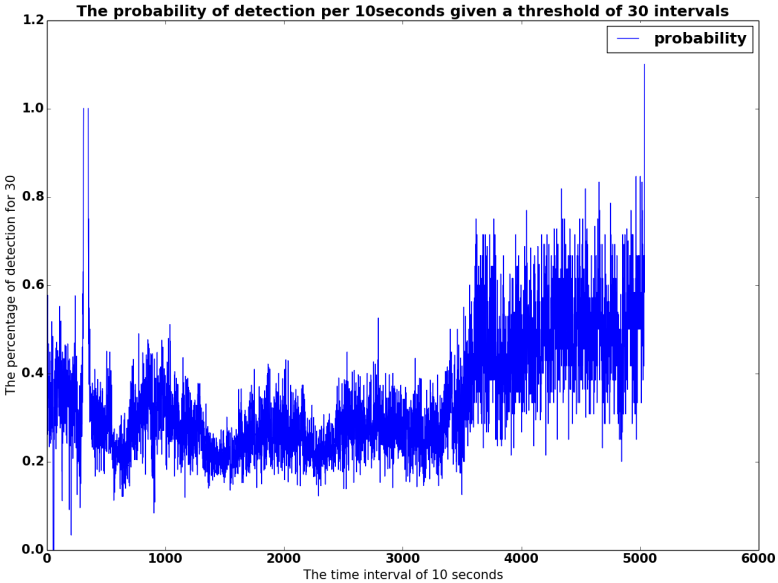
In the case of the 10 seconds time interval, the average probability is 0.337, which means that a device is detected on average 33.7% of the times between its time of its arrival and departure, while the standard deviation is 0.131. The same interpretation can be given for the 30second and 60 second time interval, which have an average probability of detection of 0.542 and 0.713, respectively. In the end, the results obtained with the correction are better than the results obtained from the basic model, which had a value of a 0.25 ± 0.09.

Another interesting aspect that can be observed in the plots of Figures 3.6a, 3.7a, and 3.8a is that the corrected ideal and the active number of detected devices have similar shapes across all three plots. The difference between them is of binning only. This represents a good result, because it reveals that the results are consistent over time, no matter the selection of the time interval. In addition to this, it can be observed in the same plots that for the the active MAC addresses and the total detected MAC addresses seem to overlap with each other, because the number of arrivals and departures per time interval is too small compared to the total detections and the number of In_out MAC addresses is zero due to the 5 record *threshold*.

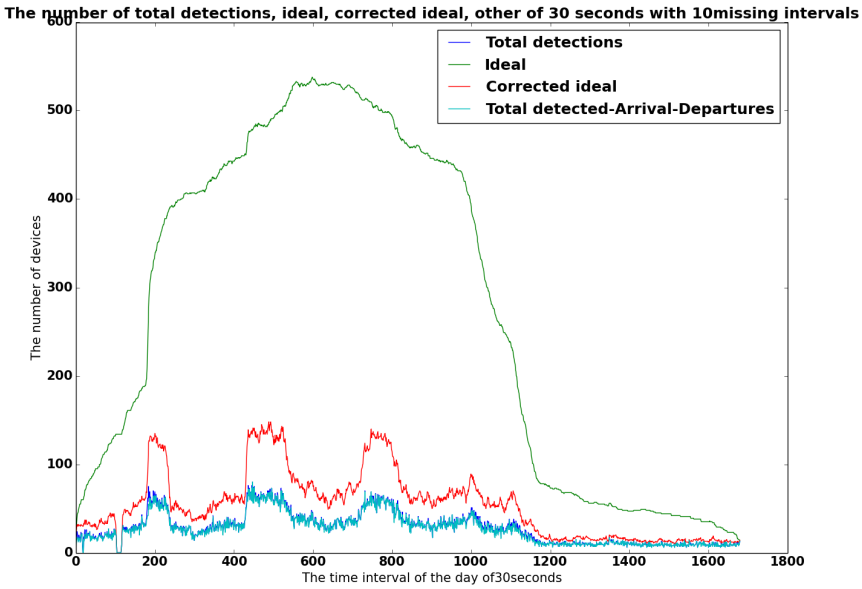
The plots show that during the day there was a failure in the beginning of the day which lasted more than 5 minutes. There were no detections during that time period and the MAC addresses were corrected. However, the ideal number of MAC addresses was not influenced by this failure, as it depends mainly on arrivals and departures. Due to the fact that the corrected ideal and the active MAC addresses were both equal to zero within that period, the probability was set to be equal to 1. Hence, these plots bring insight not only on what happens on a day level, but they also reveal information regarding possible failures of the WiFi tracking. In order to actually interpret correctly the probability of detecting mobile devices, one would also need to take into consideration the plots of the counts over day.



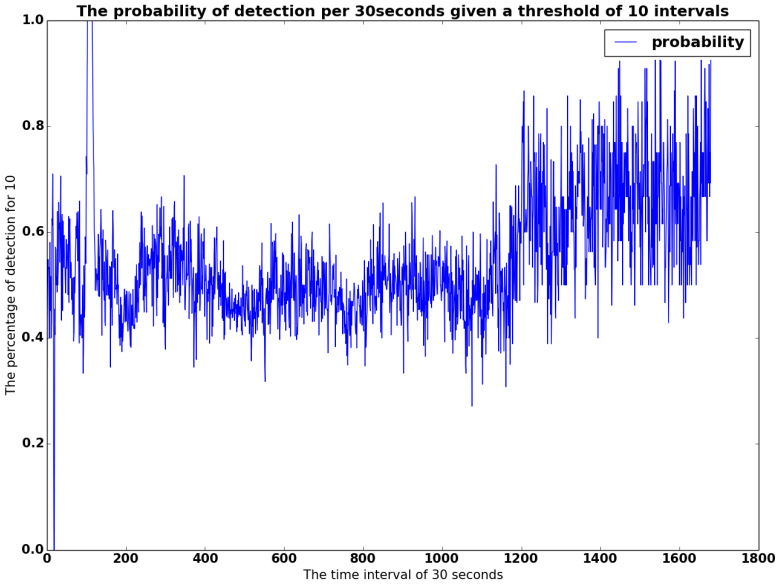
(a) The total detections, the ideal, the corrected ideal, and the active MAC addresses per 10 seconds time interval



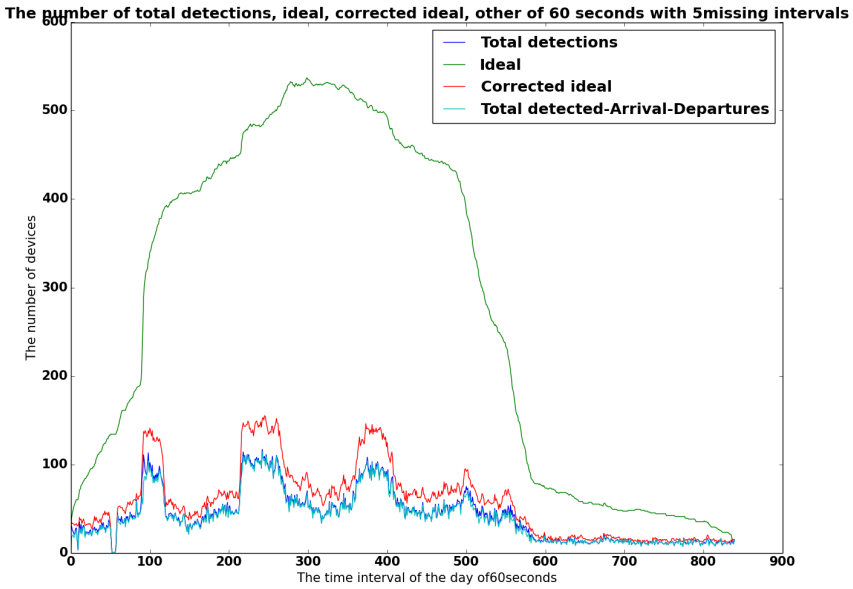
(b) The probability of detecting a mobile device per 10 time interval



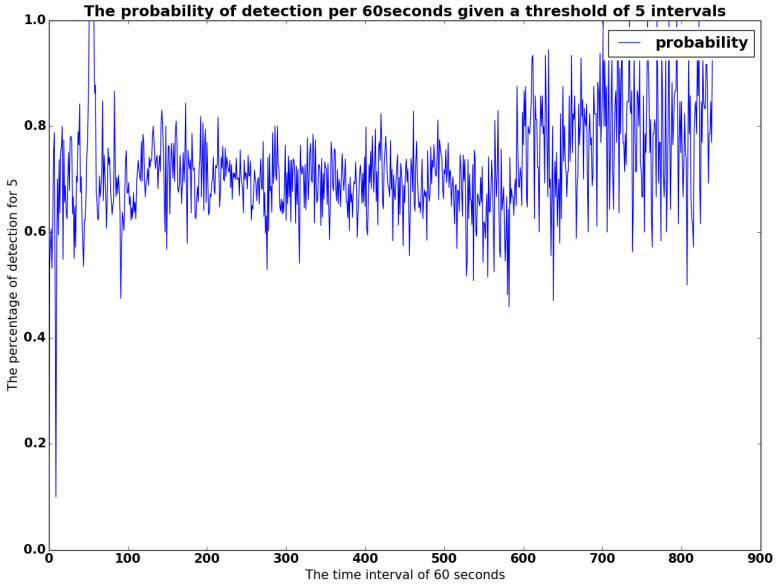
(a) The total detections, the ideal, the corrected ideal, and the active MAC addresses per 30 seconds time interval



(b) The probability of detecting a mobile device per 30 time interval



(a) The total detections, the ideal, the corrected ideal, and the active MAC addresses per 60 seconds time interval



(b) The probability of detecting a mobile device per 60 time interval

3.3 Conclusions

In this chapter, two models were proposed for calculating the probability of detecting the mobile devices: the "Basic model" and the "Cut-off model". The basic model represents an intuitive model that calculates the probability of detecting mobile devices without making any assumptions. The results showed that the probability of detecting a mobile devices is approximately 0.25 ± 0.09.

Compared to the basic model, the cut-off model cleans the dataset from MAC addresses with the number of records less than a specified threshold. It also introduces an assumption that all MAC addresses that have a $\Delta time \geq 5$ minutes are gone and it zooms-into the dataset on a 10, 30, or 60 second level instead of 5 minutes as the Basic model.

The results of this model show that the probability of detecting a mobile device increases considerably and the higher the *time_interval* is, the higher the probability of detection is.

Chapter 4

Controlled table top experiments

4.1 Introduction

In this chapter, controlled table top experiments are implemented which reveal insight in how mobile devices communicate with the routers. In addition to this, the Fitter algorithm is run on the raw datasets obtained from the experiments, as we would like to measure how accurate and frequent the detections are given the known coordinates and behaviors of the mobile devices. Furthermore, we would like to verify whether the Fitter algorithm is able to detect these devices in different circumstances.

We would like to also understand what are the most important factors that influence the detection process. According to the literature (Prasithsangaree et al.[19], Mautz[34], etc.), the most important factors that may influence the RSS(signal strength) of the Wi-Fi device as measured by the drones represent the following:

1. **The state of the device**, which can be found according to the literature in different phases like: idle, sleep, battery low, power saving mode, wireless turned on, but not connected to a network, busy using an app, streaming data, browsing, screen on and interacting with the cell-phone, screen off etc.
2. **The position of the device** that may be lied down on a table, in a pocket, in a bag, jacket, purse etc.
3. **State of the owner of the device**, which can stand still or walk. The walk can be either on a fast or slow pace and either in a circle or zig-zag.
4. **The configuration of the drones** with respect to the device that may be either on a line, triangle, etc.
5. **The packet type** which represents the type the packet the cell-phone sends. The packet types can be the following: probe, management, streaming, etc. In this situation, the probe packets are more

to be trusted than the management packets, because they are always present when the Wi-Fi is enabled.

6. **The time interval of the day** constitutes an important aspect as the presence or absence of devices may influence the RSS([19]). Thus, the devices may behave differently during busy times(morning, lunch, etc.) and less crowded time periods(after-hours,etc.).
7. **Device type** may also have a different behavior of sending packets. This can differ on the device model(different brands or operating systems), but also on the device type such as laptops, cell-phones, tablets.
8. **Environment** which can contain surrounding materials that can attenuate the signal strength, which can be found in Table 4.1

Material	**Signal attenuation**
Glass window	2db
Wooden door	3db
Cubicle	$3-5$db
Plasterboard wall	3db
Dry wall	4db
Cinderblock	5db
Marble	5db
Glass wall with metal frame	6db
Brick wall	8db
Concrete wall	$10-15$db

Table 4.1: Signal attenuation by material

4.2 Experiments

Given the time constraints, we decided to implement the following four experiments with different states of the device. Three of them have a similar setup with mobile devices found in different positions placed on the table and in different states like: active and with Wi-Fi enabled, but not connected to the internet, active using a mobile application, and idle. These states are the most frequent ones in which mobile devices are found. In addition to this, we wanted to have a first impression regarding the performance of the Fitter algorithm with respect to these factors. Another experiment was performed in order to test the assumption the algorithms use that there exists a non-linear relationship between signal strength and distance. The summary of the design and implementation of the experiments is found in Table 4.2.

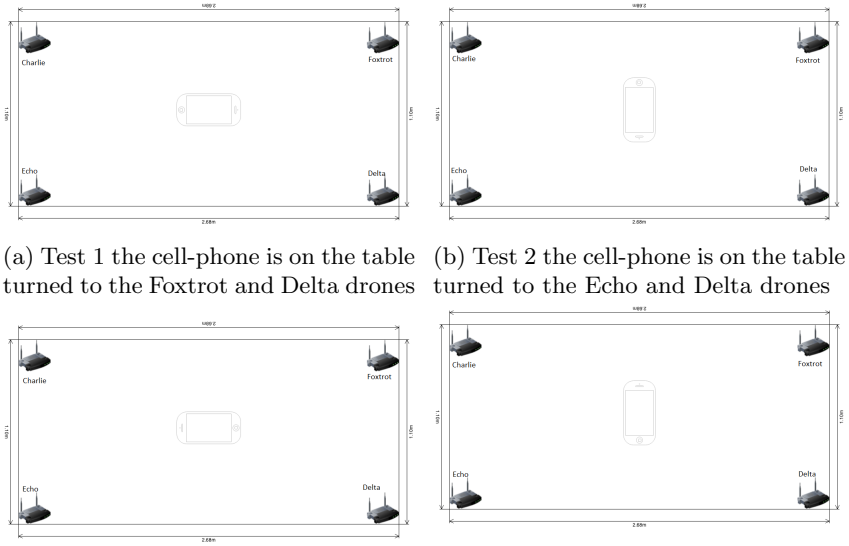
Experiment	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Device state	Wi-Fi enabled, not connected to the network	Wi-Fi enabled	Active, using a mobile application	Idle
Main packet types	probe packets	probe packets	data management	probe packets
Configuration	Charlie, Echo, Delta, Foxtrot	Charlie, Echo	Beta, Charlie, Delta, Echo	Beta, Charlie, Delta, Echo
Mobile device	iPhone	HTC desire	HTC Desire & iPhone	HTC Desire & iPhone
Device position	laid down on the table	in the hand, horizontally	laid down on the table	laid down on the table
Environment	controlled	hallway	controlled	controlled
Tests	4	3	4	2

Table 4.2: Summary of the experiments

4.2.1 Experiment 1

For this experiment, four different drones were mounted in a room. These drones were positioned at the corners of a table which had the following measurements: $length = 2.68m$ and $width = 1.10m$. The iPhone cellphone was positioned in the center of the table. The location of this center was defined with the following coordinates:(0, 0), while the corners, where the drones were positioned, were defined with following coordinates: $(-1.34m, 0.55m)$, $(-1.34m, -0.55m)$, $(1.34m, -0.55m)$, $(1.34m, 0.55m)$. For simplicity, the drones are named as follows: Charlie, Echo, Foxtrot, Delta. Besides these four drones, another drone was present which was named Alpha. Alpha was not placed in the controlled environment, but in the vicinity of the room at around 5 meters distance, because we wanted to see how Alpha detects the mobile device when it is further than the area of analysis.

Experiment 1 consists of four different tests where the iPhone was placed in four different positions, because we want to discover whether the cellphone has a directional antenna. It may the case the signal strength may be attenuated or accentuated if the antenna is located in the direction of the drones. In addition to this, we would like to check whether the drones have an individual "bias" and and whether the signal strength is constant over time given the identical position of the mobile device. Identifying this individual bias is quite important, because the coordinates are calculated based on the measured signal strength by the drones. If there is a bias of the drone, then it should be eliminated, because the accuracy of the detections will decrease. In our case, the signal strength that is received by Charlie should ideally be equal to the one received by Echo. This should also be the case for Foxtrot and Delta.



(a) Test 1 the cell-phone is on the table turned to the Foxtrot and Delta drones

(b) Test 2 the cell-phone is on the table turned to the Echo and Delta drones

(c) Test 3 the cell-phone is on the table turned to the Echo and Charlie drones

(d) Test 4 the cell-phone is on the table turned to the Charlie and Foxtrot drones

Figure 4.1: The tests of experiment 1, the iPhone is placed in the center of the table, while the Charlie, Echo, Delta, and Foxtrot drones are located at the corners of the table

Note that in this experiment:

- The drones are positioned not at the same height as the mobile device(the height is approximately $20cm$)
- The cell-phone is directly placed on a wooden table and this may attenuate the signal strength
- The drones are placed relatively close to each other(typically the drones are placed several tens of meters apart from each other)
- The Foxtrot and Delta drones are placed close to the window

For all four tests, we performed the same type of analysis. On one hand, we looked at the raw dataset, which contained the signal strengths, the drone names, the timestamps, the packet types and subtypes, out of which we only selected the probe packets which have $type = 0$ and $subtype = 4$. These types of packets represent the most reliable ones, because they will always be present when a device has Wi-Fi enabled. On the other hand, we also looked at the datasets that contain the reconstructed location of the devices. For each test, we will describe the results of both the raw dataset with the detected signal strength of each drone and the detected coordinates of the Fitter algorithm.

On an aggregated level, the number of records detected within these four experiments by each drone is visualized in Figure 4.2. It can be observed that Charlie, Delta, and Echo have the same number of records detected within all four experiments, which shows that these drones record in a

similar manner the devices. However, the Foxtrot drone has a smaller number of detections, which may indicate technical issues. In addition to this, the Alpha drone seems to have detected less records compared to the rest of the drones. An explanation for this result may be that it was in the vicinity of the room.

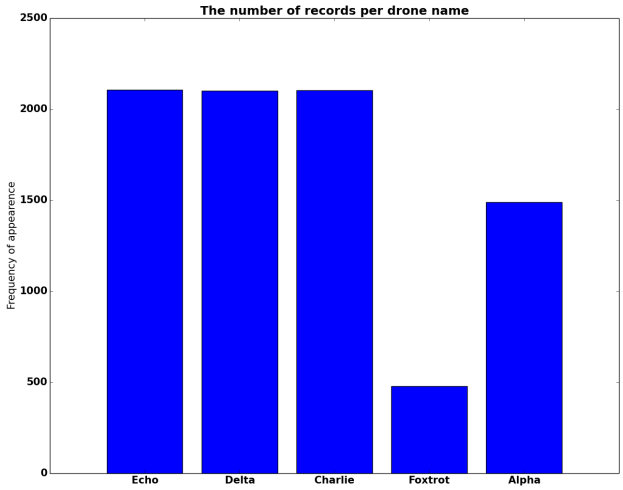


Figure 4.2: The bar plot of the number of detections per drone from all four experiments

We determined the average signal strength per drone per test. The results of these computations are found in Table 4.3 for drones Charlie, Delta, Echo, and Foxtrot drones. As it can be observed, the Foxtrot drone performs poorly and it seems to have a significantly higher average signal strength compared to the rest of the drones in all tests. The most interesting result is in Test 3, where Foxtrot has a high signal strength of 30.8 ± 3.7 compared to Delta and Echo which have the average signal strength of 46.1 ± 1.6 and 54.2 ± 2.4, respectively. Similar results were also obtained for Test 2 and Test 4. We did not analyze the Alpha drone, as it was outside the area of analysis(the room of the experiment).

Drone	Test 1	Test 2	Test 3	Test 4
Charlie(dBm)	-44.5 ± 9.7	-55.9 ± 5.2	-60.3 ± 3.1	-49.5 ± 2.8
Delta(dBm)	-52.1 ± 6.5	-48.6 ± 2.1	-46.1 ± 1.6	-49.4 ± 1.8
Echo(dBm)	-44.0 ± 9.4	-55.0 ± 5.2	-54.2 ± 2.4	-51.8 ± 2.8
Foxtrot(dBm)	-43.3 ± 9.0	-40.4 ± 5.8	-30.8 ± 3.7	-40.6 ± 5.7

Table 4.3: Average signal strength per drone and experiment

Besides the raw dataset, we also computed some average statistics for the coordinates detected by the Fitter algorithm, which can be found in

the Table 4.4. In total for all the recorded tests, there were 157 records calculated. The average values of the x coordinate are below zero for Tests 1 and 4, while for Tests 2 and 3 they are greater than zero. The average y coordinate are greater than zero for Tests 1, 3, and 4, except for Test 2 which has an average y coordinate below zero. In the end, these seem good since the average coordinates are between $[-1, 1]$, while their uncertainties are also on average below a meter.

Statistics	iPhone test1	iPhone test2	iPhone test3	iPhone test4
Records_Fitter	96	39	2	20
Records_packets	2,390	816	1,205	466
$\mu_x(m)$	-0.9	0.4	0.4	-0.2
$\mu_y(m)$	0.0	-0.1	0.3	0.0
$\sigma_x(m)$	0.6	1.0	0.7	0.8
$\sigma_y(m)$	0.3	0.7	0.2	0.8

Table 4.4: Statistics of experiment 1 iPhone

Test 1

The first test can be visualized in Figure 4.1a. As it can be seen, the iPhone device was placed with its "Home" button towards the Charlie and Echo drones. This test started at $14:55$ and ended at $15:14$. The history of the performed actions during this test can be found in Table A.1.

In Figure 4.3, the signal strength of the five drones in the first performed test of experiment 1 is plotted. It can be observed that the signal strengths of both Charlie and Echo seem to match each other, while the Delta drone is following them with a smaller signal strength. In addition to this, the Foxtrot drone has a smaller number of records compared to the other four drones, but it sometimes detects higher values compared to the other drones. As expected, the Alpha signal strength was weaker than the one received by the other drones, because it was far away from the room where the experiment was performed.

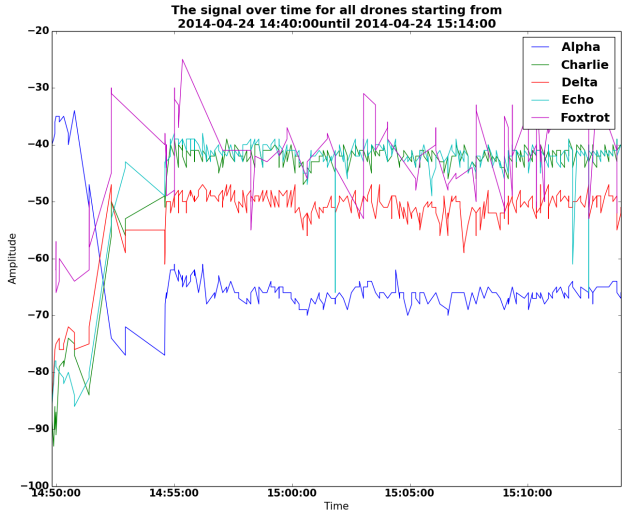


Figure 4.3: The signal strength of the five drones during the Test 1

By looking at the distribution of the signal strength, which can be found in Figure 4.4, we can observe that the distributions of Alpha, Charlie, Delta, and Echo have the same shape. However, the Foxtrot drone does not seem to follow the same distribution, which may indicate once again that this drone has some technical issues. Another interesting aspect constitutes the fact that the shapes of Charlie and Echo drones match very well. We remind the reader that these two drones were located on the same side of the table. In addition to this, according to Table 4.3, the average value of the signal strength for Charlie drone is -44.5, versus -44.0, which was computed for Echo. For the Delta drone, it can be observed that the average value of the signal strength is shifted to the left side, with a calculated value of -52.1.

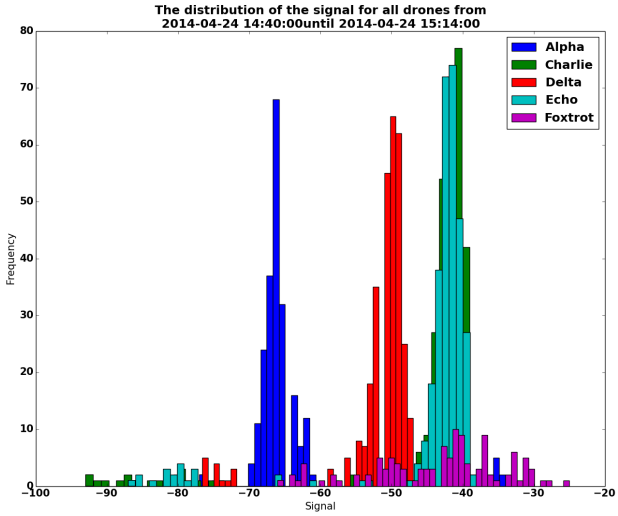


Figure 4.4: The distribution of signal strength for each of the five drones during the Test 1

When looking at the plot of the coordinates with respect to the room size and table size(Figure 4.5), the following coordinates were calculated for the position of the cell-phone. The red rectangle represents the approximate area of the room, while the green rectangle is the size of the table. In addition to this, the red dot represents the center of the table, which we defined as the point of reference, with $x = 0$ and $y = 0$. It can be seen that most of the points are centered on the table, which form a cluster. However, this cluster seems to be in the middle between the Echo and Charlie drones and the actual position of the iPhone. Another interesting aspect is that the y values of these determined coordinates are close to 0, but the x coordinates are shifted to the left side. In addition to this, there are also points which are not found on the table, but still in the room which indicate that the Fitter algorithm has a limited resolution. Even if the exact position cannot be determined precisely due to various reasons, it is still possible to detect a device within a defined space for example a room or a section of a store.

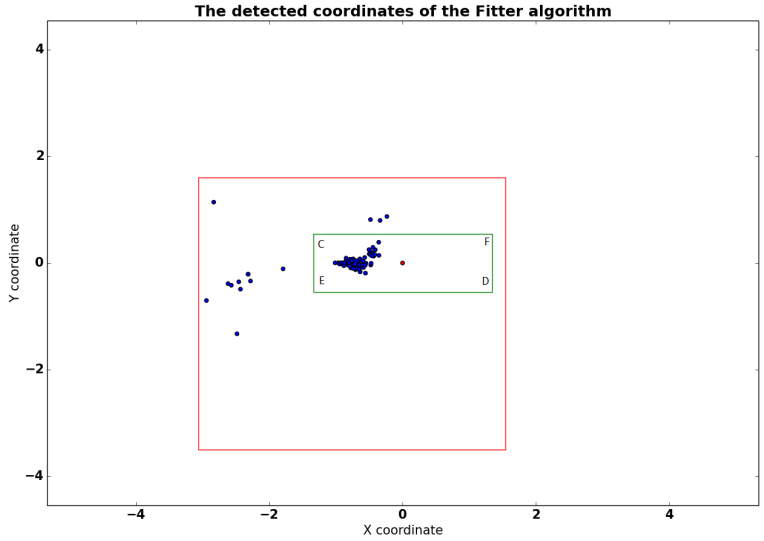


Figure 4.5: The determined coordinates by the Fitter algorithm after Test 1

The rest of the results of tests 2, 3, and 4 and their interpretation can be found in the Apendix sectionA.5, subsectionA.5.1.

Conclusions of experiment 1

After the four performed tests, we draw the following conclusions:

- The Fitter algorithm may detect the cell-phone device position in the vicinity of its actual position
- The drones behave differently when recording the signal strength
- The drones do record different signal strengths given the position of a mobile device
- No pattern could not be found with respect to the directional antenna based on the recorded signal strengths

4.2.2 Experiment 2 - testing the relationship between signal and distance

Experiment 2 was performed three times in a hallway. For this experiment, an HTC mobile device and only two drones were used: Charlie and Echo, which were mounted next to each other. The configuration of this experiment is visualized in Figure 4.6. The mobile device was set to the active mode in all three tests and there was no need to press the home button anymore like it was the case in experiment 1. For each performed test, the value of the signal strength was measured for one minute at different

distances: 3, 6, 9, 12, 15, 18, 21, 24, 27, 30 meters. The only exception was test 1 where the measurements for 30 meters were not performed by accident.

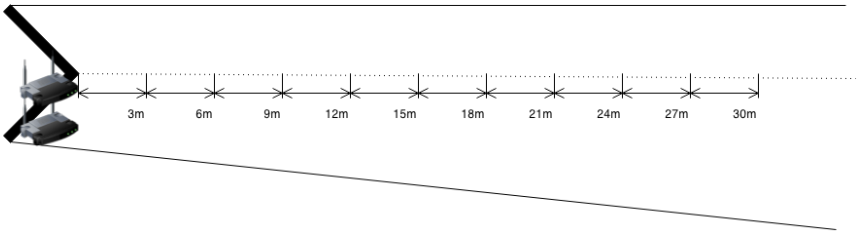


Figure 4.6: The configuration of the experiment 2

The starting and ending times of the three tests is visualized in Table A.5.

The purpose of this experiment is that it tests one of the assumptions that both algorithms developed by KPMG use, which is based on the Friis transmission equation[17], that there exists an inverse relationship between distance and the signal strength.

$$Signal\_strength = constant + 20 \times \log_{10} \frac{C}{4df\pi}$$

where f represents the frequency(fixed to 2.4GHz) and C represents the speed of light(set constant to 2.99792458units), while d represents the distance measured in meters. Given this relationship, the distance can be calculated based on which the coordinates of the position of a cell-phone can be derived using trilateration.

Thus, this experiment tries to determine whether this relationship holds the assumed one by performing three different tests which measure the signal strength for one minute at each 3 meter step. These tests contain on one hand the plots of the measurements for each 3meter interval and the detected signal strength, the chi-square minimization fit for the average of them.

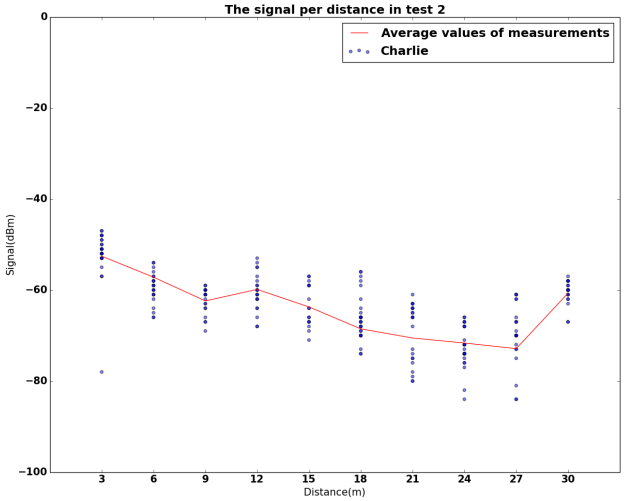
The limitations of experiment 2 are the following:

- The shape of the hallway is not straight and it contains elements(furniture, walls, metal drawers, etc.) that may attenuate the signal strength
- There were people passing by that may have influenced the way the signal strength was detected
- The measurements were taken every 3 meters

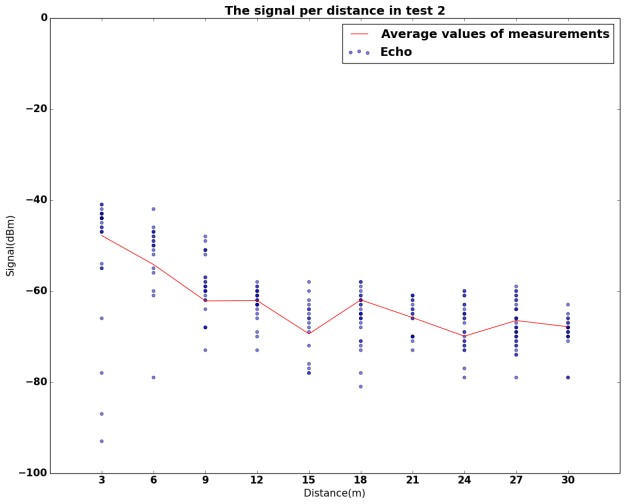
The measurements and the results of the tests 1 and 3 can be found in the Apendix subsection A.5.2. Nevertheless, we present the results of Test 2.

Measurements of the signal strength versus distance-Test 2

By looking at the measurements performed in Test 2 with the results in Figure 4.7a and Figure 4.7b, it can be seen again that the signal strength decreases the further a cell-phone device is, but this is not the case all the time. If the Charlie and Echo measurements are compared with each other, the shape of their measurements resembles, except for the 3 meter distance. In addition to this, the average value of the measurements decreases up to the 9 meters distance and then it starts increasing in both cases at the 12 meter distance. Then, on one hand, in case of the Charlie drone the average continues to decrease up to 27 meters and then it increases for the 30 meters. On the other hand, this situation does not actually happen for the Echo drone, where the average signal strength increases at the 18 meter distance, followed by another increase but less intense at 27 meters. This seems surprising as we would expect the power of the signal to decrease continuously. One possible reason for this may be the shape of the hallway, the walls, and the metal drawers which may attenuate or may increase the signal strength at different locations.



(a) Measurements of Test 2 taken by drone Charlie



(b) Measurements of Test 1 taken by drone Echo

Figure 4.7: Measurements of Test 2 from experiment 2

After recording the measurements for the mobile device, the chi-square minimization technique was used for estimating the parameters of the relationship between the signal and the distance of each performed test. Thus, the following linear relationship will be considered:

$$Signal\_strength = a - 20 \times b \times \log_{10} distance$$

where we will try to estimate a and b. The reason why 20 is included is be-

cause we would like to check the assumptions the algorithms use that the signal strengths can be fit by using a similar version to the Friis transmission equation[17]. In addition to this, we computed the uncertainties(σ_i) associated with each measurement of the signal strength.

According to Press et al.[45], if the measurement errors are normally distributed, then the chi-square merit function will give the maximum likelihood parameter estimations of a and b, which has the following formula:

$$\chi^2(a,b) = \sum_{i=1}^{N} \left(\frac{y_i - a - bx_i}{\sigma_i}\right)^2$$

This equation is minimized in order to determine a and b. At its minimum, the derivatives of $\chi^2(a,b)$ with respect to a and b will vanish. Thus, in order to perform the chi-square minimization, the $curve\_fit$ function of the *scipy.optimize* packet of Python was used for estimating both a and b parameters. This function takes as parameters:

1. **the function type**: in this case $Signal\_strength = a - 20 \times b \times \log_{10} distance$
2. **the x coordinate vector**: in this case the distance vector
3. **the y coordinate vector**: in this case the average signal strength per 3meter step
4. **the starting values of both a and b**: in this case the [-20, 1]
5. **the measurement errors vector**: in this case the standard deviation of the measurements

The results of the chi-square minimization can be visualized in Table 4.5 for Charlie and in Table 4.6. The obtained optimal parameters seem to have similar values for both a and b. We expect the a to be greater than -20dBm and b to be close to 1, because we would like this model to resemble the Friis transmission equation[17]. As it can be observed, the a parameter is significantly smaller than -20dBm in all cases, while the b parameter seems to be close to 1, especially in Test 3 of both drones. After determining the optimal parameters of both a and b, the χ^2 was calculated for each individual test, as well as $\frac{\chi^2}{dof}$, where dof represents the number of degrees of freedom. For each test, the number of the degrees of freedom is equal to 8, because it is the difference between the length of the distance vector(10) and the number of parameters(2, a and b). In addition to this, we computed the probability that a sample will be larger than $\frac{\chi^2}{dof}$. All the obtained probabilities are large for both drones, which may indicate the fact that the measurement errors may not follow a normal distribution as it was assumed. Moreover, even though this assumption may not hold, the model still seems to fit the data, as it can be seen in the next subsubsection.

Test	Charlie						
	$\hat{a}$	ε_a	$\hat{b}$	ε_b	χ^2	$\frac{\chi^2}{dof}$	p-value
$Test_1$	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.	n.d.
$Test_2$	-52.2	7.3	0.5	0.3	14.11	1.76	0.08
$Test_3$	-37.8	2.6	1.1	0.1	1.55	0.2	0.99
$Average$	-44.9	7.8	0.8	0.3	-	-	-

Table 4.5: Experiment 2 chi-square minimization Charlie drone

Test	Echo						
	$\hat{a}$	ε_a	$\hat{b}$	ε_b	χ^2	$\frac{\chi^2}{dof}$	p-value
$Test_1$	n.d.	n.d.	n.d.	n.d.	n.d.	n.d	n.d.
$Test_2$	-45.3	5.9	-0.8	0.2	3.98	0.50	0.85
$Test_3$	-35.8	3.3	-1.1	0.1	3.53	0.44	0.89
$Average$	-40.5	6.8	-0.9	0.3	-	-	-

Table 4.6: Experiment 2 chi-square minimization Echo drone

In the following subsections, the plots of the average signal strength and its lower and upper bound versus the fitted function will be presented. The fitted function was determined based on the estimated parameters of the chi-square minimization technique.

Chi-square minimization of the average signal strength test 2

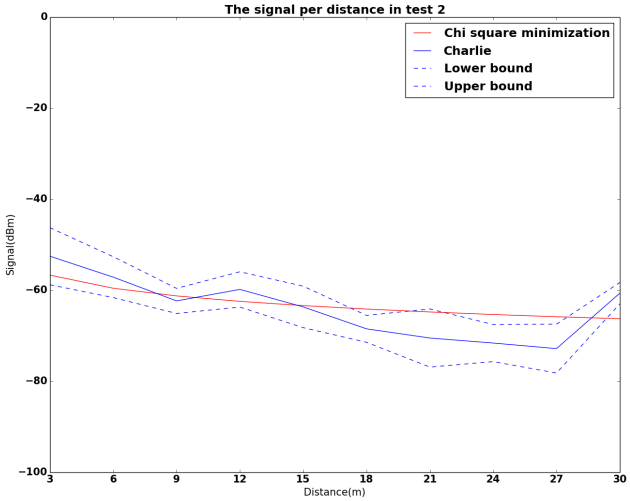
The results of these plots can be found in Figure 4.8a for the Charlie drone and in Figure 4.8b for the Echo drone. In both cases, the $curve\_fit$ returned the optimal parameters of the tests, which is:

$$f(distance) = -52.2 + 0.5 \times 20 \times \log_{10} distance$$

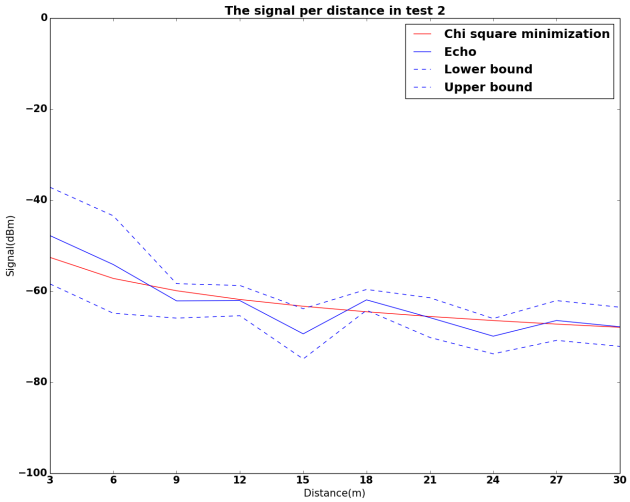
for the Charlie drone and the following for the Echo drone:

$$f(distance) = -45.3 + 0.8 \times 20 \times \log_{10} distance$$

The calculated function using these optimal parameters for Charlie drone does not seem to fit the data, as the function touches the upper bound of the average starting from the distance of 15meters. Nevertheless, this does not happen for the drone Echo, which has a calculated function that seems to follow the shape of the data.



(a) Measurements of Test 2 recorded by drone Charlie



(b) Measurements of Test 2 recorded by drone Echo

Figure 4.8: Measurements of Test 2 from experiment 2

The results of chi-square minimization of tests 1 and 3 can be found in the Appendix A.5.2.

Conclusions of experiment 2

- The drones record in a similar manner the signal strengths
- The environment under analysis plays a key role in the way the signal strengths are recorded

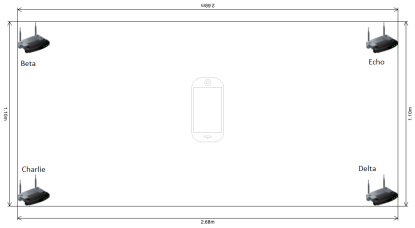
- The calculated function using the chi-square minimization seems to fit in most of the cases the measured signal strengths
- The calculated function using the chi-square minimization resembles partially the Friis equation
- It may the case that the measurement errors may not be normally distributed
- Further calibration may be needed to compensate for the environment

4.2.3 Experiment 3 - iPhone & HTC in an active state, using a mobile application

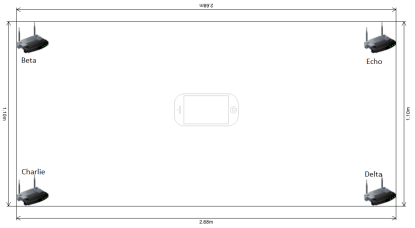
The configuration of this experiment was the same as the one mentioned in Subsection 4.2.1, where four drones were placed on the table. As seen in the results of this experiment, the Foxtrot drone behaved in a different manner than the other drones. Thus, we decided to replace this drone with another one called Beta and to verify whether the behavior is different. Thus, the names of the used drones were: Beta, Charlie, Delta, and Echo.

The motivation behind this experiment is that we would like to check whether the cell-phones communicate more and, therefore, send more packets when they are streaming data(for example: using an app to listen to music). In addition to this, it would be interesting to see whether it has an impact on the calculation of the coordinates, e.g., having better coordinates than the ones obtained in experiment1.

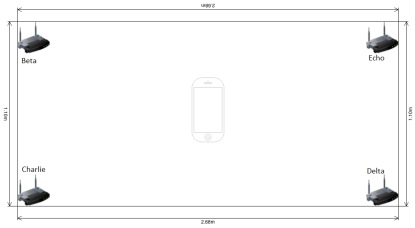
For this experiment, we used two different mobile devices: HTC and iPhone, which we tested by changing their position in the center of the table in four different directions. Each test was performed for 16 minutes where the phones were not touched. For simplicity, the plots of the positions can be visualized in Figure 4.9. One of the reasons why we chose to switch the position of the mobile device is that we wanted to check whether there is a detection bias of the drone and whether the mobile antenna has a different signal strength when the position is modified.



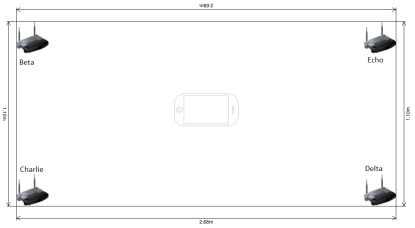
(a) Test 1 the cell-phone is on the table turned to the Echo and Delta drones



(b) Test 2 the cell-phone is on the table turned to the Chalie and Delta drones



(c) Test 3 the cell-phone is on the table turned to the Beta and Charlie drones



(d) Test 4 the cell-phone is on the table turned to the Beta and Echo drones

Figure 4.9: The tests of experiment 3, the HTC and iPhone are each placed in the center of the table, while the Beta, Charlie, Delta, and Echo drones are located in the corners of the table

The starting and ending times of these tests can be visualized in the Appendix in Table A.6.

The limitations of this experiment are the following:

- Only one main position is tested(the mobile device laid on the table)
- The drones are located close to each other
- The experiments last only 16 minutes
- The *Delta* and *Echo* drones are placed close to the window

The number of packets stored in the database for each test and drone can be visualized in Table 4.7.

Experiment		Beta	Charlie	Delta	Echo
Experiment 3-HTC	Test 1	15586	15551	15529	15619
	Test 2	15290	15487	15745	15752
	Test 3	18060	18150	17897	18005
	Test 4	14589	14482	14065	14267
Experiment 3-iPhone	Test 1	23978	23152	23402	23668
	Test 2	22882	22949	22822	22788
	Test 3	22829	22503	22648	22601
	Test 4	23567	23446	23178	23304
Experiment 4	HTC	901	899	867	898
	iPhone	820	794	777	776

Table 4.7: Number of packets per experiment, test, and drone

Experiment		Beta		Charlie		Delta		Echo	
		μ(dBm)	σ(dBm)	μ(dBm)	σ(dBm)	μ(dBm)	σ(dBm)	μ(dBm)	σ(dBm)
HTC	T_1	-49.8	1.6	-49.8	1.4	-37.2	1.4	-49.4	1.2
	T_2	-59.9	0.9	-56.9	0.9	-37.9	1.6	-52.1	1.2
	T_3	-54.7	2.7	-47.7	2.6	-38.9	2.1	-46.5	2.8
	T_4	-47.6	2.3	-45.9	1.6	-46.5	2.9	-52.1	2.8
iPhone	T_1	-42.9	1.4	-57.7	1.0	-42.1	1.4	-48.6	1.3
	T_2	-51.4	1.8	-41.2	1.1	-31.6	1.2	-46.1	1.2
	T_3	-46.7	1.4	-51.8	1.4	-35.9	1.4	-49.1	1.2
	T_4	-42.8	1.4	-41.1	1.1	-37.5	1.4	-41.9	1.0

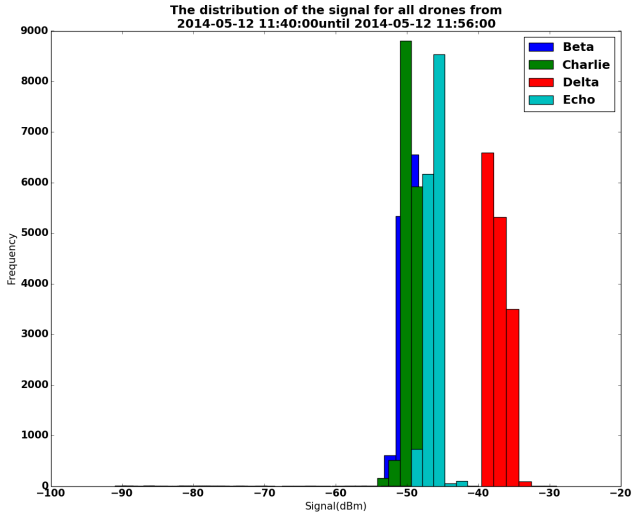
Table 4.8: The signal strength per experiment, test, and drone

Experiment		# records	X coordinate(m)		Y coordinate(m)	
			μ(m)	σ(m)	μ(m)	σ(m)
HTC	T_1	42	0.9	0.0	-0.8	0.1
	T_2	10	1.3	0.7	-0.7	0.1
	T_3	25	0.9	0.1	-0.6	0.2
	T_4	63	-0.1	0.3	-0.6	0.2
iPhone	T_1	67	0.7	0.3	-0.1	0.2
	T_2	0	-	-	-	-
	T_3	2	0.4	0.4	0.1	0.3
	T_4	93	0.3	0.0	-0.9	0.1

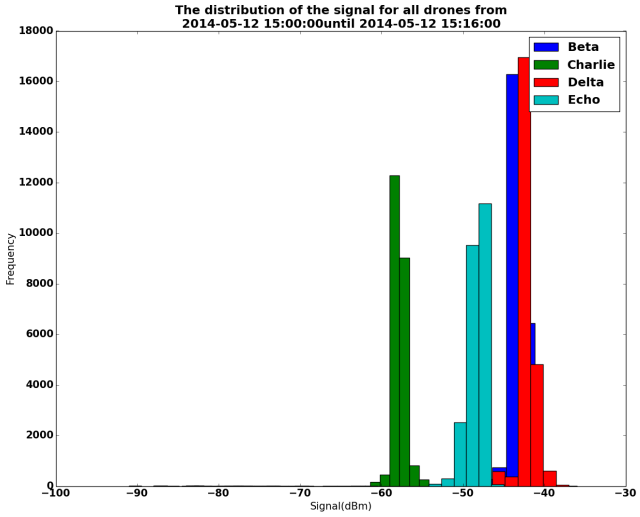
Table 4.9: The statistics of the Fitter algorithm per coordinate

4.2.3.1 Test 1

The configuration of this experiment can be found in Figure 4.9a. In order to understand the behavior of the drones, we decided to plot the distributions of the signal strength for each of them. By looking at Figures 4.10a and 4.10b, the shapes of all signal strength distributions resemble the normal distribution for all drones. In addition to this, the Delta drone has the highest average signal strength in both cases with -37.2 ± 1.4dBm(HTC) and -42.1 ± 1.4dBm(iPhone), respectively. On one hand, in the HTC case it can be seen that this average is by far higher than the other ones. One the other hand, this is not the case for the iPhone, where the average of the Beta drone is very close to Delta's(-42.9 ± 1.4dBm). Moreover, it can be visualized in Figure 4.10a that Beta and Charlie have similar values on average which is an interesting result given that these two drones are located next to each other.



(a) HTC mobile device



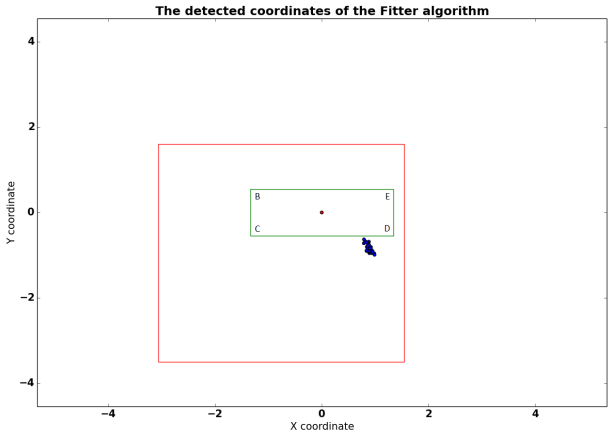
(b) iPhone mobile device

Figure 4.10: The distributions of the signal strength of Test 1 from experiment 3

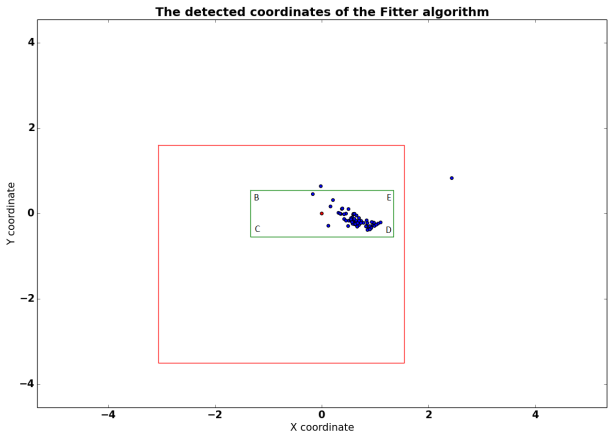
The Figures 4.11 show the calculated coordinates that result from the Fitter algorithm. According to Table 4.9, for this particular test, there were 42 points calculated for the HTC and 71 for the iPhone, respectively. In addition to this, the average x-coordinate was 0.9 ± 0.1 compared to 0.7 ± 0.3 in the iPhone's case, while the average y-coordinate was -0.8 ± 0.1 for the y coordinate in the HTC case and -0.1 ± 0.2.

These results seem good particularly for the y coordinate, which is very

close to its reference value of 0. Another interesting result is that the points are clustered and most of the points are located in the area of the room and, in the iPhone's case, most of them are actually on the table but shifted to the right side of the table and close to the Delta drone. One particular explanation for this may be that the highest average signal had been recorded for the Delta drone. As mentioned previously, the Fitter algorithm gives a higher weight to a higher signal strength, which may actually influence the determined values of the coordinates.



(a) HTC mobile device



(b) iPhone mobile device

Figure 4.11: The coordinates detected by the Fitter algorithm in case of Test 1 from experiment 3

The rest of the tests together with their results are found in the Appendix A.5.3.

4.2.3.2 Conclusions-experiment 3

After performing all these tests of experiment 3, the following conclusions can be drawn:

- Cell-phones communicate differently with the drones(e.g., they send more packets which include different signal strength values from drone to drone)
- Different drones in the identical conditions detect the same device with different signal strengths
- The computed coordinates of the phone may be influenced and placed towards a drone which receives packets with higher signal strength
- The average drone signal strength differs with device type and device position
- The Fitter algorithm does not construct all time the time the coordinates of mobile devices and further investigation needs to be done
- The average calculated coordinates have values in general in the $[-1, 1]$
- In the case of the iPhone, there were less points calculated for this experiment compared to the HTC mobile device

4.2.4 Experiment 4-Idle state of the HTC and iPhone devices

The fourth experiment was performed in the same configuration as the previous one with the Beta, Charlie, Delta, and Echo drones, which can be visualized in Figure 4.12. However, both cell-phone devices were in an idle state and laid on the table next to each other. This experiment was recorded for one hour.

The motivation behind experiment 4 is that we would like to verify how the mobile devices send packets when they are idle and whether the algorithms can actually determine accurate coordinates based on this raw dataset, how often they can do that, and whether there is a big difference between cell-phones. We chose not to change the position of the cell-phones, because we believed that this may not actually influence the way the packets are sent and because of time limitations.

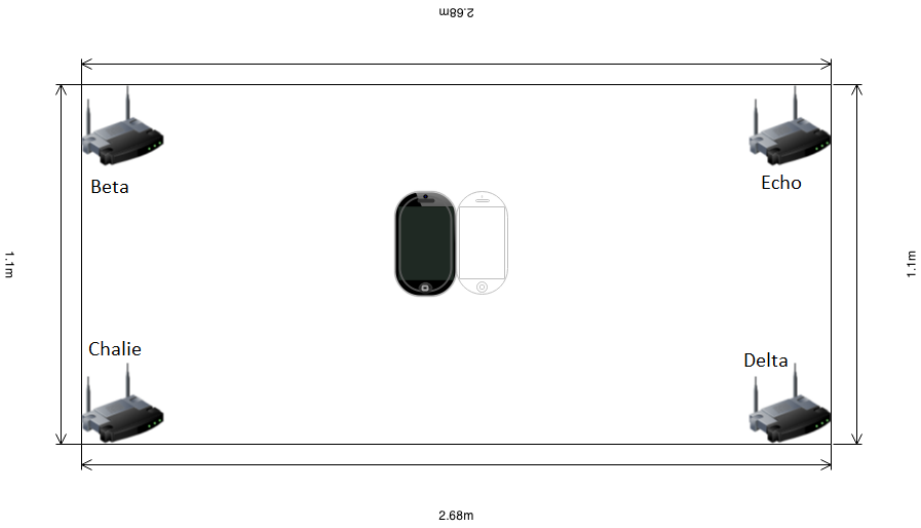


Figure 4.12: The configuration of the experiment 4 where the iPhone was placed next to the HTC

The limitations of this experiment are the following:

- There was only one position tested for this experiment
- The mobile devices were tested both at the same time
- The test only took one hour
- The mobile devices were placed on a table lower than the actual drones
- The phones contain different apps(that run in the background and influence the data traffic)

As shown in Table 4.10, there are more packets recorded by the drones for the HTC than for the iPhone mobile device when the cell phones are idle. This result is in contrast to the one obtained in experiment 3, where the iPhone communicated more that the HTC with the drones with streaming data(see Table 4.7).

Experiment		Beta	Charlie	Delta	Echo
Experiment 4	HTC	901	899	867	898
	iPhone	820	794	777	776

Table 4.10: Number of packets per experiment, test, and drone

According to Table 4.11, on one hand, the highest average signal strength was recorded for the Delta drone in case of the iPhone(-38.7 ± 5.6), while for the HTC device on average the highest signal strength was detected by Charlie(-41.3 ± 2.4). This may mean that the cell-phones have a preference towards a certain drone. In addition to this, the distributions of the

signal strength obtained in this test resemble the ones obtained in the forth test of experiment 3 performed for both the HTC and the iPhone. However, there are several differences between these two experiments. Firstly, there is a time recording difference: the test 4 of experiment 3 lasted 16 minutes, while this test lasted one hour. Secondly, both devices are in another state: the active state(listening to music on an app) versus the idleness state. Lastly, it can be clearly seen a high difference between the number of records obtained by the idleness test which are fewer than the results obtained by the data management packets. On the other hand, the distributions of the rest of the drones(Beta, Delta, and Echo) overlap with each other.

Experiment	Beta		Charlie		Delta		Echo	
	μ(dBm)	σ(dBm)	μ(dBm)	σ(dBm)	μ(dBm)	σ(dBm)	μ(dBm)	σ(dBm)
HTC	-48.3	2.6	-41.3	2.4	-48.1	1.2	-49.3	1.9
iPhone	-48.8	6.8	-44.5	6.3	-38.7	5.6	-48.9	4.4

Table 4.11: Average signal strength per device and drone experiment 4

Experiment	# Records	X coordinate		Y coordinate	
		μ(m)	σ(m)	μ(m)	σ(m)
HTC	21	-0.5	0.1	-0.9	0.2
iPhone	1	0.2	0.0	-0.2	0.0

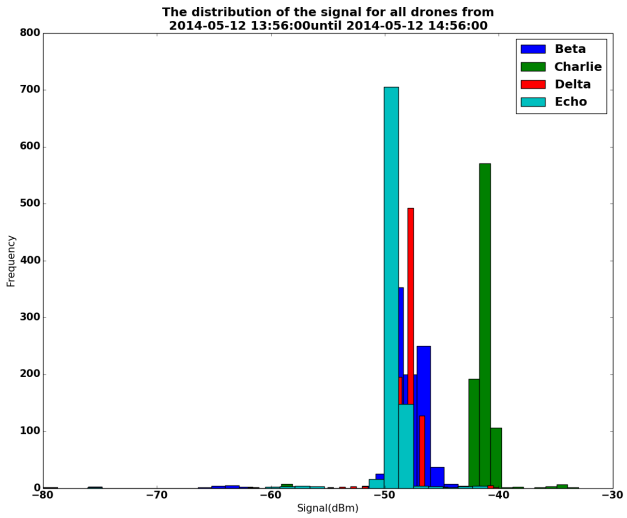
Table 4.12: Average coordinates Fitter algorithm experiment 4

As shown in to Table 4.12, there were 21 points(HTC) and 1 point(iPhone), respectively reconstructed by the Fitter algorithm for the idleness experiments. In the HTC case, the average value of the x-coordinate is -0.5 ± 0.1, while for the y-coordinate is -0.9 ± 0.2, which represent good results given the state of the device. A negative aspect is that the algorithm was not able to actually reconstruct more than one data point in case of the iPhone device even though there was an increased number of "detected" packets.

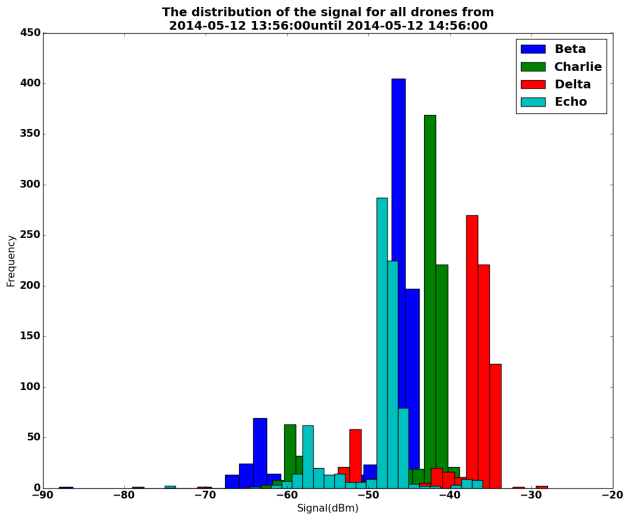
When looking at the drones' signal distribution, which can be found in Figure 4.13a in the HTC case, it can be seen that on average the Charlie drone has higher signal values which is interesting as this situation did not happen previously in other experiments. Compared to the other experiments, the distributions of the signals of each drones do not have the nice normal distribution shapes as previously seen. In Figure 4.13b, the double peaks can be seen in the iPhone case, where the Delta drone still has on average the highest signal strength, while the other drones seem to coincide with each other. Another particularity that is visualized in both Figure 4.13a and Figure 4.13b is that on average the values of Echo and Beta coincide with each other.

Even though the direction of the cell-phones was identical to the one found in experiment 3, test 4, but with the two mobile devices located next to each other, both the average values of the signal strength and the

uncertainties and the signal strength distributions do not coincide with the ones obtained with this idleness experiment. This might mean that the position of the mobile device may impact the way the signal strength is detected.



(a) The distribution of signal of the HTC mobile device in experiment 4

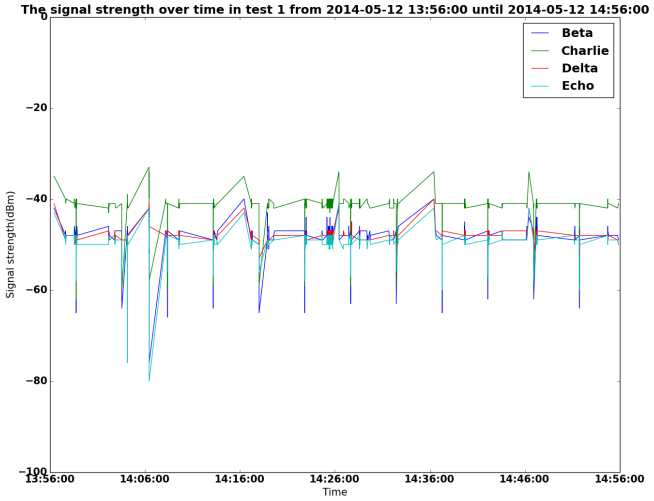


(b) The distribution of signal of the iPhone mobile device in experiment 4

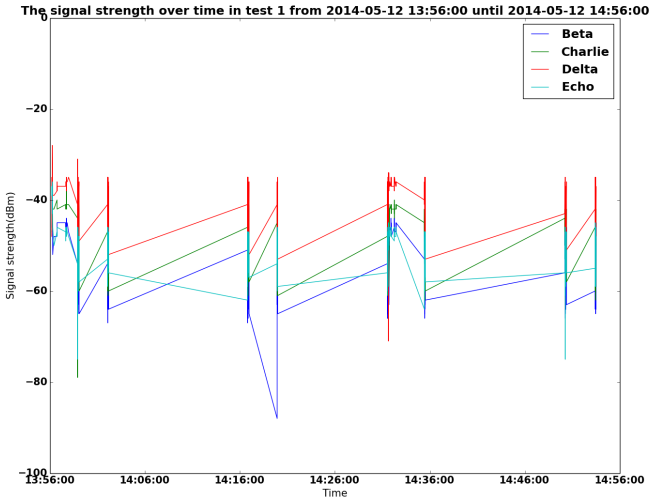
Figure 4.13: The distributions of the signal strength from experiment 4

Figures 4.14a and 4.14b reveal information about the frequency and am-

plitude of the signal over time. Here, we are able to see the time gaps between packets being sent. Moreover, if there are no packets detected at a particular time, these plots reveal this information as well as whether there is a pattern in which mobile cell-phone communicate with the drones. There is clearly a different behavior in time between two device types. In the HTC case, we see a consistent behavior over time which is much more frequent than in the iPhone case. This is interesting as in the future experiments based on the history of the signal strength and data modeling may be done which may predict the mobile device type. This information may answer the question related to whether a mobile device is in the interest area, even though the mobile device did not communicate too much.



(a) The distribution of signal of the HTC mobile device in experiment 4 over time

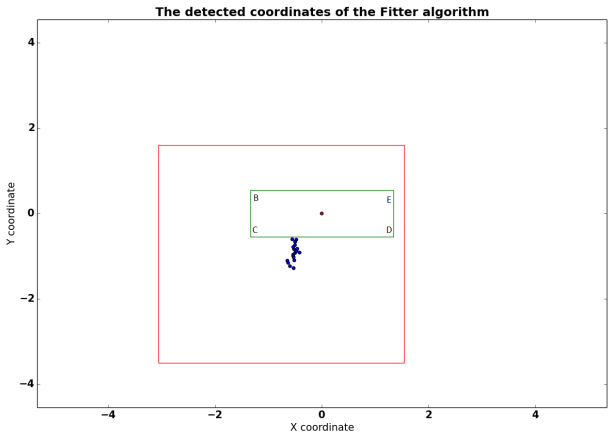


(b) The distribution of signal of the iPhone mobile device in experiment 4 over time

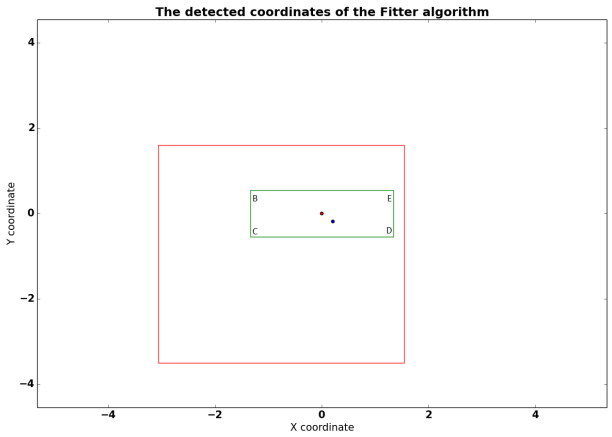
Figure 4.14: The distributions of the signal strength from experiment 4 over time

Figures 4.15a and 4.15b show the reconstructed position/coordinates of the mobile devices calculated by the Fitter algorithm. In the HTC case, it can be seen that there are far more points clustered outside the table area. As expected, the position of these points is close to the Charlie drone, as this was the one which detected on average the highest signal

strength compared to the rest of the drones. In the iPhone case, it can be seen that there is only one point reconstructed by the algorithm. This is a surprising result as there are sufficient packets. Apparently, the Fitter algorithm was not able to properly reconstruct other points and it needs further investigation.



(a) The coordinates detected by the Fitter algorithm for the HTC mobile device in experiment 4



(b) The coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 4

Figure 4.15: The coordinates detected by the Fitter algorithm for experiment 4

4.2.4.1 Conclusions-experiment 4

- The idle mobile phones communicate far less with the drones compared to the ones in active state(experiment 1 and experiment 3)
- The number of points reconstructed in the iPhone case is much smaller than in the HTC case
- The Fitter algorithm did not reconstruct too many coordinates in this experiment, even though there are more than 4 data packets detected by the drones

Chapter 5

The performance of the Fitter algorithm

5.1 Introduction

In the previous chapter, we observed that the Fitter algorithm did not reconstruct the coordinates of the cell-phone in most cases even though there was a sufficient number of packets(more than 4). In this chapter, we decided to run the analysis of counting packets versus the states for the experiments with data points(experiment 1, experiment 3, and experiment 4), because we did not observe a consistent outcome of the Fitter across all of these experiments. A similar algorithm to the first part of the cut-off model was used, which computes the behavior of a device. The difference is that the behavior is used for counting purposes for both the coordinates and the packets. The results showed that there was a bug in the algorithm, which was later fixed.

The comparison between the coordinates and the raw packets contains several steps. Firstly, the number of packets per 10 second interval is computed, as well as the number of points calculated by the algorithm. Secondly, a binary variable called *state* is introduced in the case of the coordinates for every 10 seconds time interval. The *state* is equal to one, if there were coordinates calculated for that particular device and time interval, and zero otherwise. Lastly, for each time interval of 10 seconds, the number of packets and the state are compared by means of a graphical representation. This graphical representation can indicate which intervals have detections or not and packets or not.

For a better understanding of what actually happened, we decided to analyze in parallel both the previous datasets and the current obtained datasets by looking at the following elements: the quantities that assess the performance of the Fitter algorithm and the indicators regarding the efficiency of the Fitter algorithm and of the WiFi tracking system, respectively.

5.2 Quantities that assess the performance of the Fitter algorithm

Several quantities were computed for each test and experiment in order to evaluate the performance of the Fitter algorithm before and after the bug fix, which have a 0 index if they refer to the results prior to the bug fix(before), while 1 indexes refer to the results after the bug fix(after).

There are two types of quantities:

- related to the total number of packets(**Records_packets**)
- related to the number of reconstructed data points by the Fitter (**Records_Fitter**)

Three different categories were created for the comparison between data packets and states which have as a measurement unit a number of 10 seconds time intervals. The categories are the following:

- ni_{Fitter} are the number of time intervals that have both points reconstructed by the algorithm and the number of data packets not equal to zero(**state** $\neq$ **0 and Records_packets** $\neq$ **0**).
- ni_0 are the number of intervals with no reconstructed points by the Fitter, but with records different than zero in the raw dataset(**state** = **0 and Records_packets** $\neq$ **0**).
- $ni_{nopackets}$ are the number of intervals with neither records in the raw dataset nor records reconstructed by the Fitter algorithm(**state** = **0 and Records_packets** = **0**).

The sum of these should be equal to the the maximum number of time intervals for a particular test, the **intervals** variable, which represents the number of time intervals of size 10 seconds that are between the arrival and departure of a device or the start and end time of an experiment.

$$intervals = ni_{Fitter} + ni_0 + ni_{nopackets}$$

The arrivals and departures of the WiFi devices are quantified as follows:

- **Arr_interval_pack**, the number of the first time interval when the mobile device sent packets to the routers and the $time_{start}$.
- **Dep_interval_pack** represents the number of remaining intervals between the $time_{stop}$ and the last interval when the mobile device sent packets to the routers
- **Arr_interval_alg**, the number of intervals between the interval of first point detected by the algorithm and the $time_{start}$
- **Dep_interval_alg**, the number of intervals between the $time_{stop}$ and the time of the last point detected by the algorithm

Other important quantities that are relevant for the performance are related to the resolution of the Fitter algorithm:

- μ_x represents the average value of the x-coordinate for a particular test and experiment
- μ_y represents the average value of the y-coordinate for a particular test and experiment
- σ_x represents the root mean squared error of the x-coordinate for a particular test and experiment
- σ_y represents the root mean squared error the y-coordinate for a particular test and experiment

Eventually, the tracking system performance was evaluated quantitatively based on the indicators below, which refer to the performance of the algorithm and to the performance of the WiFi tracking system.

Prob_ddeted_alg represents the probability that the algorithm computes the points for a selected time interval of for example 10 seconds. Moreover, it is the fraction of the number of intervals with coordinates computed by the Fitter algorithm(ni_{Fitter}) divided by the number of time intervals with data packets.

$$Prob\_ddeted\_alg = \frac{ni_{Fitter}}{ni_{Fitter} + ni_0}$$

Efficiency_wifi represents the probability that the WiFi tracking system communicates during a time interval(e.g., 10 seconds). These statistics are calculated within the entire measurement period [$time_{start}$, $time_{stop}$]. [1]

$$Efficiency\_wifi = \frac{ni_{Fitter} + ni_0}{ni_{Fitter} + ni_0 + ni_{nopackets}}$$

Efficiency_wifi_ui constitutes the probability that the WiFi tracking system communicates every selected time interval(e.g., 10 seconds) with the mobile device within the time interval [*Arrival*, *Departure*], where the *Arrival* represents the time interval when the first detection of a particular device was seen first, while *Departure* represents the time interval where the last detection of that particular devices was seen. $Arr\_interval\_alg$ represents the number of 10 seconds intervals of the difference between the *Arrival* and $time_{start}$, while $Dep\_interval\_alg$ represents the number of 10 seconds intervals of the difference between the $time_{stop}$ and the *Departure*. The $Efficiency\_wifi$ and $Efficiency\_wifi\_ui$ differ when the *Arrival* and *Departure* of a WiFi device do not correspond to $time_{start}$ and $time_{stop}$.1^0

$$Efficiency\_wifi\_ui = \frac{ni_{Fitter} + ni_0}{ni_{Fitter} + ni_0 + ni_{nopackets} - Arr\_interval\_alg - Dep\_interval\_alg}$$

[1] The reason why this interval is important is because sometimes the analyst is not able to choose the exact moment when the first packets or coordinates were calculated by the algorithm.

Total_efficiency represents the probability of reconstructing the coordinates of the mobile devices given the performance of the WiFi tracking system for every selected time interval(e.g., 10 seconds).

$$Total\_efficiency = Prob\_ddeted\_alg \times Efficiency\_wifi$$

Total_efficiency_ui is the probability of the Fitter reconstructing the coordinates for the mobile devices given its efficiency in the interval [$Arrival$, $Departure$]. This indicator should be compared with the $Total\_efficiency$ indicator.

$$Total\_efficiency\_ui = Prob\_ddeted\_alg \times Efficiency\_wifi\_ui$$

5.3 Experiment 1

It is reminded that for this experiment, we tested the communication between mobile devices and routers when the cell-phones are not connected to the network. In this state, the mobile devices send probe packets.

In Table 5.1, a difference can be observed between the results before and after the bug fix. Initially, the algorithm computed a smaller number of records for each test, but after the algorithm was corrected the number increased(which is a good result). For example, the highest increase was in Test 3 of this experiment, which initially had only 2 points calculated, but after fixing the bug there were 81 points added. As it can be seen in the same table, the number of intervals without packets is similar for all four tests except for test 2 which contains 3 time intervals of 10second compared to the rest which contain only 2 time intervals of 10 seconds.

When looking at the probability that the algorithm computes records(this was completely different from test to test prior the bug fix), but it increased afterwards to intervals of probability with the upper bound of 100% for all four tests. Due to the fact that the arrival time and the time start of the tests coincide, the $Efficiency\_wifi$ and $Efficiency\_wifi\_ui$ coincide. It is interesting to see that within this experiment almost all 10 seconds contain data packets.

The $Total\_efficiency$ and $Total\_efficiency\_ui$ are different before and after the bug fix as they depend on the probability that the algorithm calculates points.

Statistics	iPhone test1	iPhone test2	iPhone test3	iPhone test4
$Records\_Fitter_0$	96	39	2	20
$Records\_Fitter_1$	113	88	83	59
Records_packets	2,390	816	1,205	466
$ni_{Fitter0}$	96	39	2	20
$ni_{Fitter1}$	113	88	83	59
$ni_{nostates0}$	17	49	81	39
$ni_{nostates1}$	0	0	0	0
$ni_{nopackets0}$	2	3	2	1
$ni_{nopackets1}$	2	3	2	1
Intervals	115	91	85	60
$Prob\_ddeted\_alg_0$	$84.9\% \pm 11.6\%$	$44.3\% \pm 18.1\%$	$2.4\% \pm 5.8\%$	$33.9\% \pm 21.2\%$
$Prob\_ddeted\_alg_1$	$[99\%, 100\%]$	$[98.7\%, 100\%]$	$[98.6\%, 100\%]$	$[98.1\%, 100\%]$
$Efficiency\_wifi_0$	$98.3\% \pm 4.2\%$	$96.7\% \pm 6.5\%$	$97.6\% \pm 5.8\%$	$98.3\% \pm 5.8\%$
$Efficiency\_wifi_1$	$98.3\% \pm 4.2\%$	$96.7\% \pm 6.5\%$	$97.6\% \pm 5.8\%$	$98.3\% \pm 5.8\%$
$Efficiency\_wifi\_ui_0$	$98.3\% \pm 4.2\%$	$96.7\% \pm 6.5\%$	$97.6\% \pm 5.8\%$	$98.3\% \pm 5.8\%$
$Efficiency\_wifi\_ui_1$	$98.3\% \pm 4.2\%$	$96.7\% \pm 6.5\%$	$97.6\% \pm 5.8\%$	$98.3\% \pm 5.8\%$
$Total\_efficiency_0$	$83.5\% \pm 12.0\%$	$42.9\% \pm 18.0\%$	$2.4\% \pm 5.8\%$	$33.3\% \pm 21.2\%$
$Total\_efficiency_1$	$98.3\% \pm 4.2\%$	$96.7\% \pm 6.5\%$	$97.6\% \pm 5.8\%$	$98.3\% \pm 5.8\%$
$Total\_efficiency\_ui_0$	$83.5\% \pm 12.0\%$	$42.9\% \pm 18.0\%$	$2.4\% \pm 5.8\%$	$33.3\% \pm 21.2\%$
$Total\_efficiency\_ui_1$	$98.3\% \pm 4.2\%$	$96.7\% \pm 6.5\%$	$97.6\% \pm 5.8\%$	$98.3\% \pm 5.8\%$
$\mu_{x0}(m)$	-0.9	0.4	0.4	-0.2
$\mu_{x1}(m)$	-1.1	-1.3	-71.6	0.5
$\mu_{y0}(m)$	0.0	-0.1	0.3	0.0
$\mu_{y1}(m)$	0.2	-1.2	-5.1	0.7
$\sigma_{x0}(m)$	0.6	1.0	0.7	0.8
$\sigma_{x1}(m)$	0.9	2.0	501.4	0.8
$\sigma_{y0}(m)$	0.3	0.7	0.2	0.8
$\sigma_{y1}(m)$	0.8	3.7	41.2	1.5

Table 5.1: Statistics of experiment 1 iPhone(before(0) and after(1) fix results)

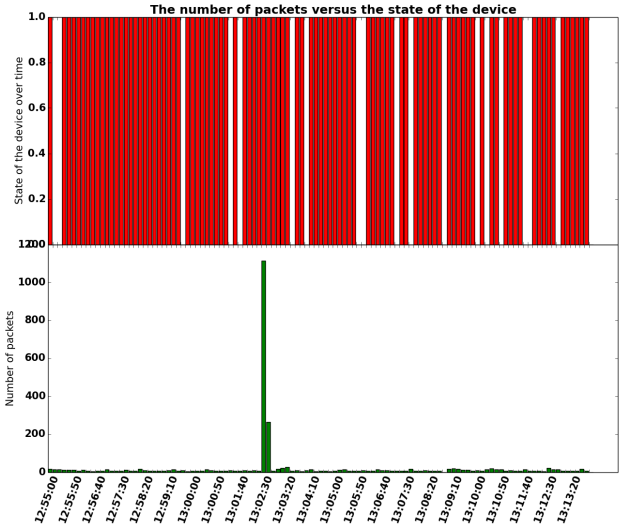
5.3.1 Test 1

In the Figure 5.1, the comparison between the states and the data packets can be seen and the 10 seconds time intervals which have detections or not. Figure 5.1a shows whether there were detections for this particular before the bug fix, while Figure 5.1b shows the improvement. The later figure reveals that two time intervals of 10 second have no packets, while the rest of the intervals all have detections. Test 1 represented the test with the largest number of points calculated by algorithm before the bug fix, but, after the fix, the 17 more points were detected.

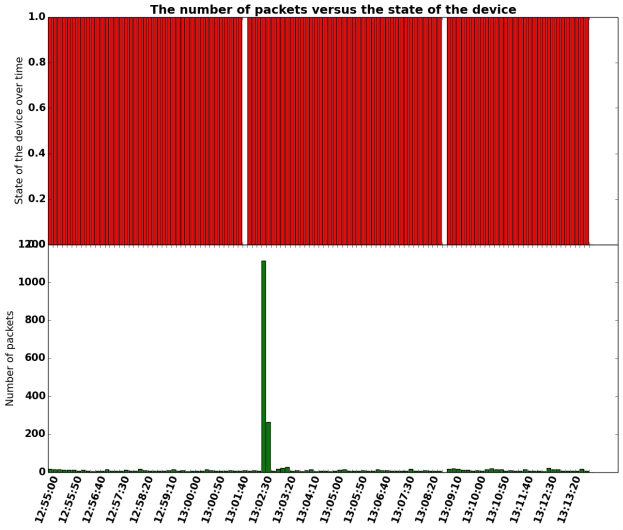
According to the Table 5.1, even though there were much more points the performance of the average x and y coordinates decreased. Initially, the average x coordinate was -0.9 ± 0.6 and the average y coordinate was 0.0 ± 0.3 for this test. However, after the correction both values had values further away from 0, the reference point for both coordinates, and their spreads(σ_x and σ_y increased as well). Hence, there was an increase

in detection efficiency, but this increase did not improve the precision in resolution.

When looking at Figure 5.2, the coordinates calculated by the algorithm are plotted. The red rectangle represents the area of the room, while the green rectangle represents the area of the table on which the mobile device was placed. It can be seen that the points detected before the bug fix are still present in the plots after the fix. Figure 5.2b reveals additional points, which are in the area of the room, but are not located on the table. This is in accordance with the fact that there were improvements from a number of data points calculated, but this did not bring any increase the accuracy of the detections for both x and y. One can see 3 main areas with multiple data points. On one hand, there is a cluster of points that located on the table, but shifted to the left side of the center of reference with the y coordinate close to 0. On the other hand, there are two other areas with points in the room, but not on the table. These points are either located in the vicinity of Charlie and Echo routers, but with their y coordinate close to 0 or in the middle between the Charlie and Foxtrot routers, but with their x coordinate close to 0. According to Table 4.3, the Foxtrot, Echo, and Charlie routers had the highest signal strength. The algorithm uses this information and fits the signals according to their magnitude. Thus, the calculated points will be closer to the router with a higher signal strength. The orientation of the mobile device might not have a large impact on the reconstruction of the coordinates compared to the recorded signal strength.

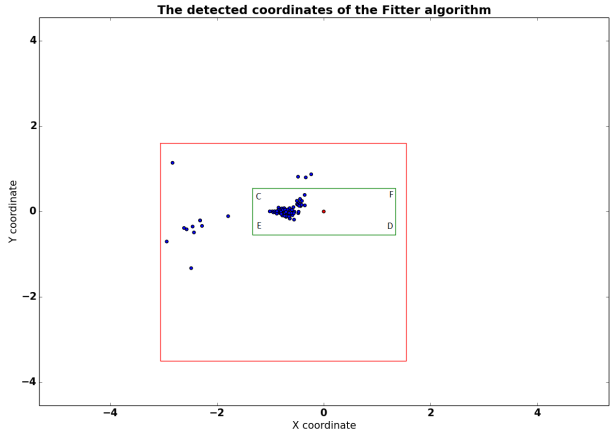


(a) The comparison between the states and data packets for experiment 1 Test 1(before)

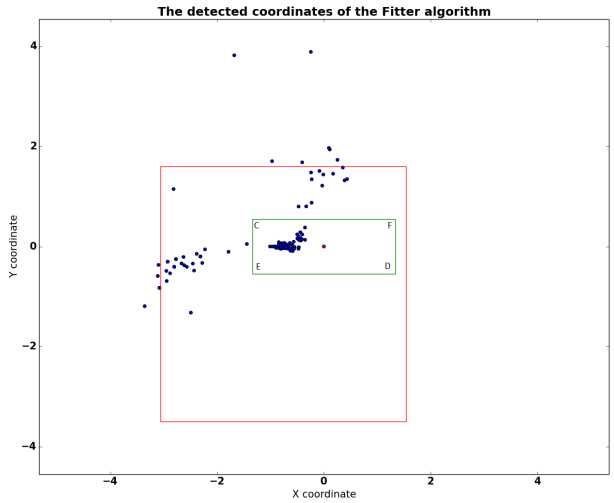


(b) The comparison between the states and data packets for experiment 1 Test 1(after)

Figure 5.1: The comparison between the states and data packets for experiment 1 Test 1 before and after



(a) The calculated coordinates by the Fitter algorithm experiment 1 Test 1 (before)



(b) The calculated coordinates by the Fitter algorithm experiment 1 Test 1 (after)

Figure 5.2: The calculated coordinates by the Fitter algorithm experiment 1 Test 1(before and after)

The rest of the tests and the results of their analyses can be found in the Appendix in section A.6

5.4 Experiment 3

We remind the reader that experiment 3 was performed in order to analyze the way the mobile devices communicate when an app is used. There

were two main mobile devices used HTC and iPhone, which were tested under the same conditions and for the same amount of time. From a number of packets perspective, according to Tables 5.2 and 5.3, the HTC communicates less with the routers than the iPhone which can be observed given the number of recorded packets per test for each device. Therefore, a higher number of points was expected to be computed in the iPhone case. However, this only occurred for tests 1 and 4. Moreover, for tests 2 and 3, the HTC contained much more points than the iPhone.

After the Fitter algorithm was fixed, the number of calculated points was similar except for test 3, where for the iPhone device were computed 95 points compared to the 94 number of points calculated for the HTC. This result is reassuring as this may indicate that the device type may not influence the number of points calculated by the algorithm. In addition to this, the total efficiency of the tracking system is very high for both devices. It can be also seen that the average values of the x and y coordinates did not significantly change. This is a positive result compared to the values obtained in experiment 1, where the resolution deprecated.

Statistics	HTC test1	HTC test2	HTC test3	HTC test4
Records_packets	62,285	62,274	72,112	57,403
$ni_{Fitter0}$	42	10	25	63
$ni_{Fitter1}$	94	95	94	94
$ni_{nostates0}$	54	86	72	33
$ni_{nostates1}$	2	1	2	2
$ni_{nopackets0}$	1	1	1	1
$ni_{nopackets1}$	1	1	1	1
Intervals	97	97	97	97
$Prob\_ddeted\_alg_0$	$43.8\% \pm 17.5\%$	$10.4\% \pm 10.8\%$	$26\% \pm 15.5\%$	$65.6\% \pm 16.8\%$
$Prob\_ddeted\_alg_1$	$97.9\% \pm 5.1\%$	$99.0\% \pm 3.5\%$	$97.9\% \pm 5.1\%$	$97.9\% \pm 5.1\%$
$Efficiency\_wifi_0$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$
$Efficiency\_wifi_1$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$	$99\% \pm 3.5\%$	$99.0\% \pm 3.5\%$
$Efficiency\_wifi\_ui_0$	$[98.2\%, 100\%]$	$[98.2\%, 100\%]$	$[98.2\%, 100\%]$	$[98.2\%, 100\%]$
$Efficiency\_wifi\_ui_1$	$[98.2\%, 100\%]$	$[98.2\%, 100\%]$	$[98.2\%, 100\%]$	$[98.2\%, 100\%]$
$Total\_efficiency_0$	$43.3\% \pm 17.5\%$	$10.3\% \pm 10.7\%$	$25.7\% \pm 15.4\%$	$64.9\% \pm 16.8\%$
$Total\_efficiency_1$	$96.9\% \pm 6.1\%$	$97.9\% \pm 5.1\%$	$96.9\% \pm 6.1\%$	$96.9\% \pm 6.1\%$
$Total\_efficiency\_ui_0$	$43.8\% \pm 17.5\%$	$10.4\% \pm 10.8\%$	$26\% \pm 15.5\%$	$65.6\% \pm 16.8\%$
$Total\_efficiency\_ui_1$	$97.9\% \pm 5.1\%$	$99.0\% \pm 3.5\%$	$97.9\% \pm 5.1\%$	$97.9\% \pm 5.1\%$
$\mu_{x0}(m)$	0.9	1.3	0.9	-0.1
$\mu_{x1}(m)$	1.0	1.3	0.9	-0.2
$\mu_{y0}(m)$	-0.8	-0.7	-0.6	-0.6
$\mu_{y1}(m)$	-1.0	-0.8	-1	-0.9
$\sigma_{x0}(m)$	0.0	0.1	0.1	0.3
$\sigma_{x1}(m)$	0.1	0.1	0.1	0.3
$\sigma_{y0}(m)$	0.1	0.1	0.2	0.2
$\sigma_{y1}(m)$	0.1	0.0	0.2	0.5

Table 5.2: Statistics of experiment 3 HTC(before(0) and after(1) fix results)

Statistics	iPhone test1	iPhone test2	iPhone test3	iPhone test4
Records_packets	94,200	91,441	90,581	93,495
$ni_{Fitter0}$	67	0	2	93
$ni_{Fitter1}$	94	95	95	94
$ni_{nostates0}$	29	0	94	3
$ni_{nostates1}$	2	1	1	2
$ni_{nopackets0}$	1	1	1	1
$ni_{nopackets1}$	1	1	1	1
Intervals	97	97	97	97
$Prob\_ddeted\_alg_0$	$69.8\% \pm 16.2\%$	$[0\%, 1.2\%]$	$2.08\% \pm 5.0\%$	$96.9\% \pm 6.1\%$
$Prob\_ddeted\_alg_1$	$97.9\% \pm 5.1\%$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$	$97.9\% \pm 5.1\%$
$Efficiency\_wifi_0$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$
$Efficiency\_wifi_1$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$
$Efficiency\_wifi\_ui_0$	$[98.8\%, 100\%]$	$[98.8\%, 100\%]$	$[98.8\%, 100\%]$	$[98.8\%, 100\%]$
$Efficiency\_wifi\_ui_1$	$[98.8\%, 100\%]$	$[98.8\%, 100\%]$	$[98.8\%, 100\%]$	$[98.8\%, 100\%]$
$Total\_efficiency_0$	$69.1\% \pm 16.3\%$	$[0\%, 1.2\%]$	$2.1\% \pm 5.1\%$	$95.9\% \pm 7.0\%$
$Total\_efficiency_1$	$96.9\% \pm 6.1\%$	$97.9\% \pm 5.1\%$	$96.9\% \pm 6.1\%$	$96.9\% \pm 6.1\%$
$Total\_efficiency\_ui_0$	$69.8\% \pm 16.2\%$	$[0\%, 1.2\%]$	$2.1\% \pm 5.1\%$	$96.9\% \pm 6.1\%$
$Total\_efficiency\_ui_1$	$97.9\% \pm 5.1\%$	$99.0\% \pm 3.5\%$	$99.0\% \pm 3.5\%$	$97.9\% \pm 5.1\%$
$\mu_{x0}(m)$	0.7	-	0.4	0.3
$\mu_{x1}(m)$	0.4	1.2	1.0	0.3
$\mu_{y0}(m)$	-0.1	-	0.1	-0.9
$\mu_{y1}(m)$	0.4	-1.2	-1.0	-0.9
$\sigma_{x0}(m)$	0.3	-	0.4	0.0
$\sigma_{x1}(m)$	0.5	2.2	0.2	0.0
$\sigma_{y0}(m)$	0.2	-	0.3	0.1
$\sigma_{y1}(m)$	0.8	1.0	0.3	0.2

Table 5.3: Statistics of experiment 3 iPhone(before(0) and after(1) fix results)

5.4.1 Test 1

In this test, there was a significant difference from a number of computed points perspective. Thus, as previously seen, where there were packets the algorithm was able to compute points for the majority of the 10 time intervals which contained data packets, except for 2 for both mobile devices. Besides this, it could be seen that there existed a time interval which did not contain both packets and calculated points. This can be visualized in Figure 5.3 for the HTC case and in Figure 5.5 for the iPhone.

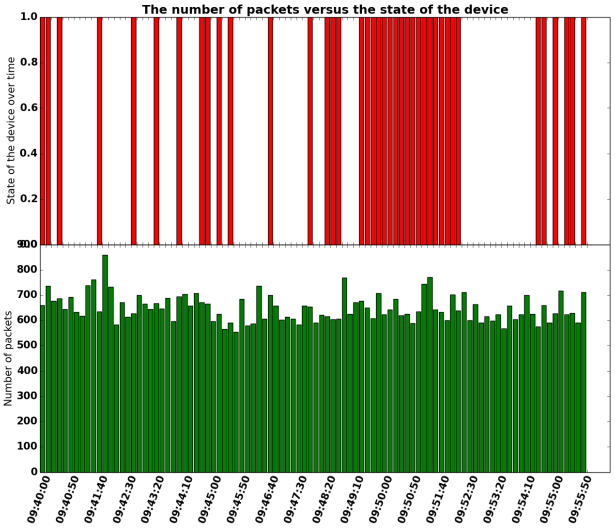
5.4.1.1 HTC

The plots of the coordinates(Figure 5.4) show that a cluster of points was formed in the vicinity of router Delta. According to the results of experiment 3 Test 1(Subsubsection 4.2.3.1), the Delta delta drone had on average the highest signal strength. Thus, one potential explanation for the location of the points calculated by the algorithm is that the high average signal strength of recorded by this drone which seems to influence the algorithm.

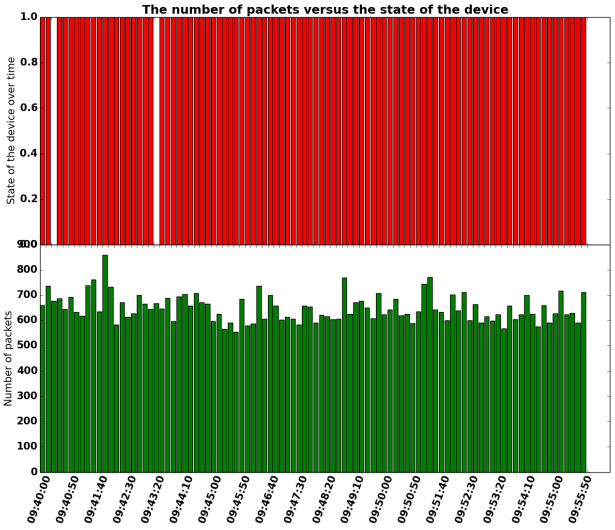
Another interesting aspect can be seen in Figures 5.3a and 5.3b, where there seems to be no significant difference between the results obtained prior to the bug fix and current ones besides the increased number of points obtained in the later. However, the majority of these points are located outside the area of the table, but still in the room.

5.4.1.2 iPhone

In Test 1 of the iPhone, a similar situation as in the HTC case occurs, where the main cluster of points was maintained after the algorithm was fix(Figure 5.6). This cluster contains as well points that are close to the Delta drone as in the HTC case, but these points are located closer to the reference point for the y coordinate. Besides the main cluster, there can be also points that are located both outside the area of the table and the area of the room. These are outliers, which seem to have the x coordinate close to 0. One potential explanation may be that the distribution of the signal strength of the Beta router overlapped with the one obtained for the Delta drone(Subsubsection 4.2.1) and may have influenced the calculated coordinates.

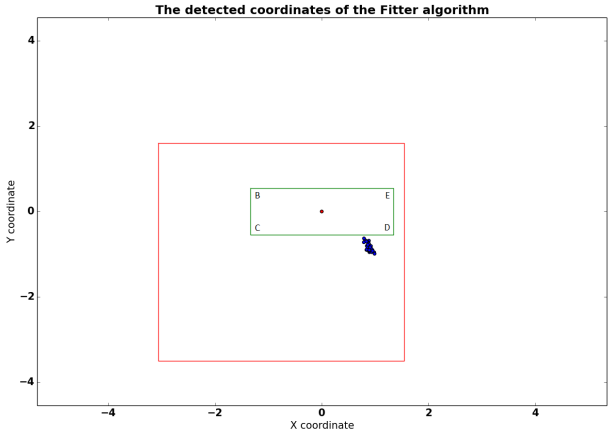


(a) The states and coordinates detected by the Fitter algorithm for the HTC mobile device experiment 3 Test 1(before)

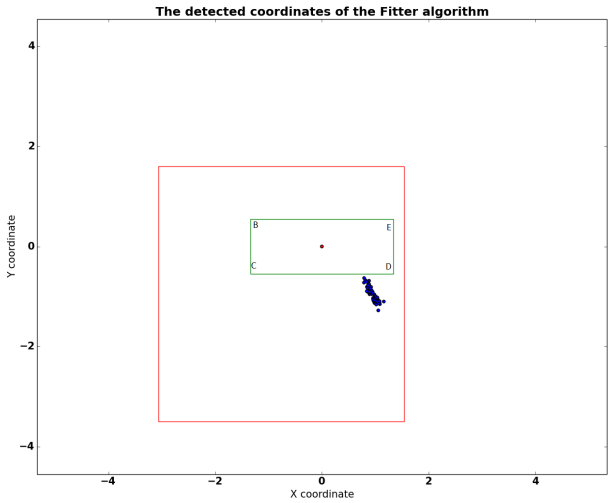


(b) The states and coordinates detected by the Fitter algorithm for the HTC mobile device experiment 3 Test 1(after)

Figure 5.3: The states and coordinates detected by the Fitter algorithm for the HTC mobile device experiment 3 Test 1

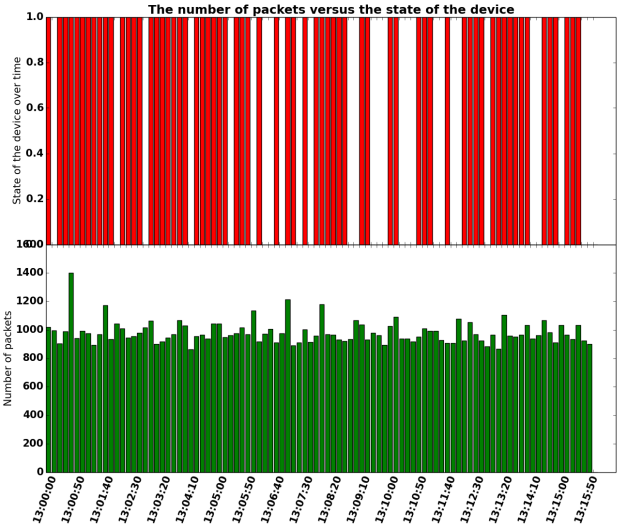


(a) The calculated coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 1(before)

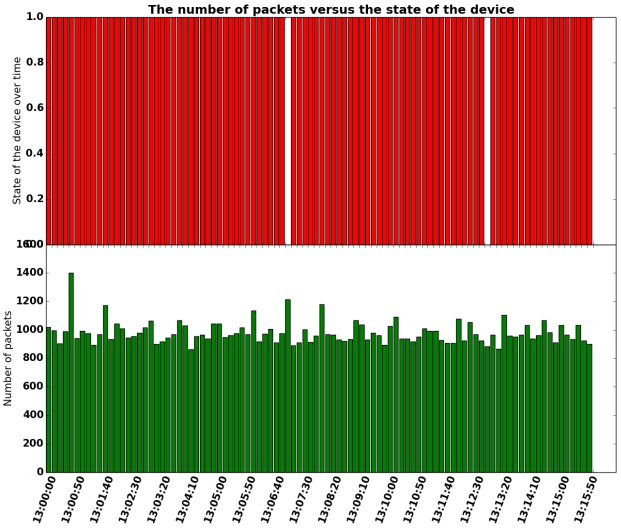


(b) The calculated coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 1(after)

Figure 5.4: The calculated coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 1(before and after)

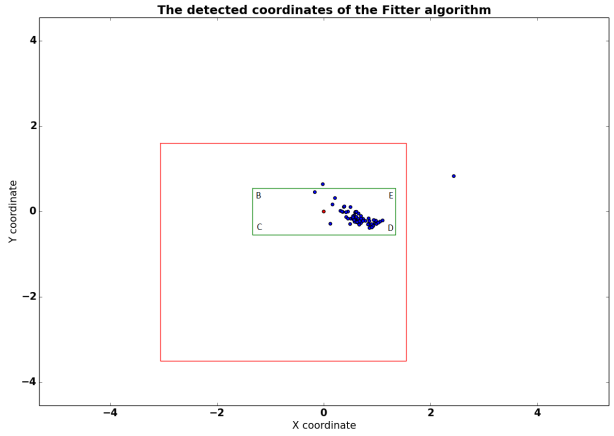


(a) The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 1(before)

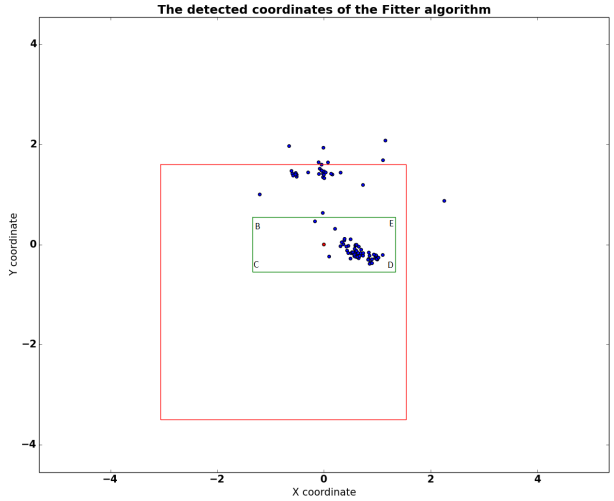


(b) The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 1(after)

Figure 5.5: The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 1(before and after)



(a) The calculated coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 1(before)



(b) The calculated coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 1(after)

Figure 5.6: The calculated coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 1(before and after)

The rest of the tests and the results of their analyses can be found in the Appendix in section A.7

5.5 Experiment 4

Experiment 4 was performed for both the HTC and iPhone at the same time in order to get insight in how mobile devices communicate with the

routers when they are in an idle state. This experiment was important, because the idle state is the most common state of a mobile device. Interesting information could be drawn out of this experiment. Initially, the algorithm could barely compute points for the mobile devices. Thus, according to Table 5.4, the initial probability was $30.9\% \pm 16.3\%$ for the HTC compared to $4.6\% \pm 7.4\%$ for the iPhone, these results changed to the intervals of probability $[98.4\%, 100\%]$ and $[95.1\%, 100\%]$, respectively, after the algorithm was fixed. Nevertheless, the efficiency of the tracking system scored initially $18.8\% \pm 13.8\%$ for the HTC mobile device and only $6.1\% \pm 8.4$ for the iPhone. This seems to be a poor result compared to experiment 1 and 3 which had data packets in the majority of the 10 seconds time intervals. This may mean that the mobile devices send fewer packets and tracking seems a challenge when they are in an idle state.

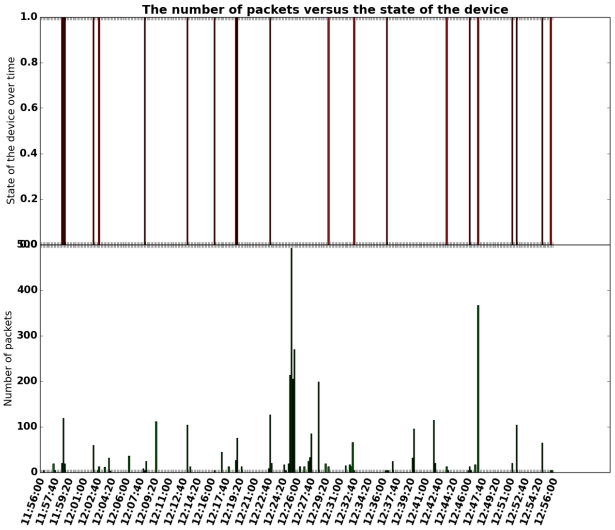
The main difference could be seen from a number of points computed for both mobile devices. According to Table 5.4, there were in the end 68 points for the HTC and 22 points calculated for the iPhone. These constituted points for all the intervals which contained data packets. From a coordinate perspective, the results after the algorithm was fixed revealed that the average x and y coordinates of the HTC mobile device did not differ compared to the ones initially obtained, but they were slightly increased. However, in the iPhone case, there was a significant change as 21 points more were calculated. For example, the average x coordinate was 0.6 ± 0.2, while the average y coordinate was -1.3 ± 0.3.

Statistics	HTC	iPhone
Records_packets	3565	3167
$ni_{Fitter0}$	21	1
$ni_{Fitter1}$	68	22
$ni_{nostates0}$	47	21
$ni_{nostates1}$	0	0
$ni_{nopackets0}$	293	339
$ni_{nopackets1}$	293	339
Intervals	361	361
$Prob\_ddeted\_alg_0$	$30.9\% \pm 16.3\%$	$4.6\% \pm 7.4\%$
$Prob\_ddeted\_alg_1$	$[98.4\%, 100\%]$	$[95.1\%, 100\%]$
$Efficiency\_wifi_0$	$18.8\% \pm 13.8\%$	$6.1\% \pm 8.4\%$
$Efficiency\_wifi_1$	$18.8\% \pm 13.8\%$	$6.1\% \pm 8.4\%$
$Efficiency\_wifi\_ui_0$	$18.9\% \pm 13.8\%$	$6.4\% \pm 8.6\%$
$Efficiency\_wifi\_ui_1$	$6.1\% \pm 8.4\%$	$18.8\% \pm 13.8\%$
$Total\_efficiency_0$	$18.8\% \pm 13.8\%$	$0.3\% \pm 1.9\%$
$Total\_efficiency_1$	$18.8\% \pm 13.8\%$	$6.1\% \pm 8.4\%$
$Total\_efficiency\_ui_0$	$5.9\% \pm 13.8\%$	$0.3\% \pm 1.9\%$
$Total\_efficiency\_ui_1$	$18.9\% \pm 13.8\%$	$6.4\% \pm 8.6\%$
$\mu_{x0}(m)$	-0.5	0.2
$\mu_{x1}(m)$	-0.6	0.6
$\mu_{y0}(m)$	-0.9	-0.2
$\mu_{y1}(m)$	-1.3	-1.3
$\sigma_{x0}(m)$	0.1	0.0
$\sigma_{x1}(m)$	0.2	0.2
$\sigma_{y0}(m)$	0.2	0.0
$\sigma_{y1}(m)$	0.3	0.3

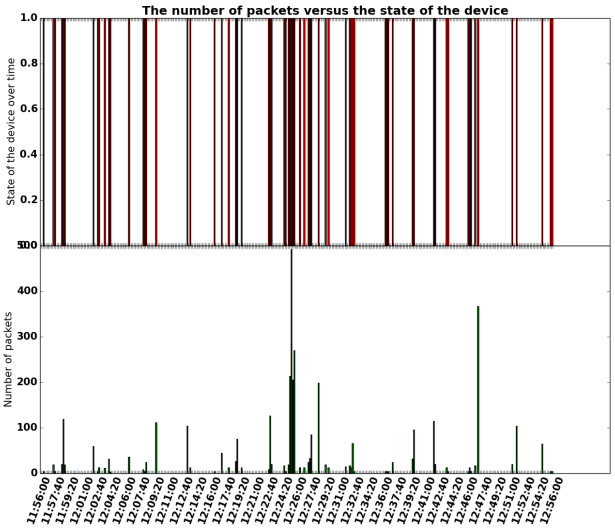
Table 5.4: Statistics of experiment 4 HTC and iPhone (before(0) and after(1) fix results)

5.5.1 HTC

According to Figure 5.8, it can be seen that the initial cluster of points was increased after the algorithm was repaired. In addition to this, the position of the cluster is between the Charlie and Delta routers, on the left side of the reference point, very close to the area of the table and inside the room. In addition to this, the additional points seem to be more scattered than the initial ones. One potential explanation for this shift to the left side may be that the Charlie router had on average the highest signal strength, followed by the Delta and Beta(very close to Delta).

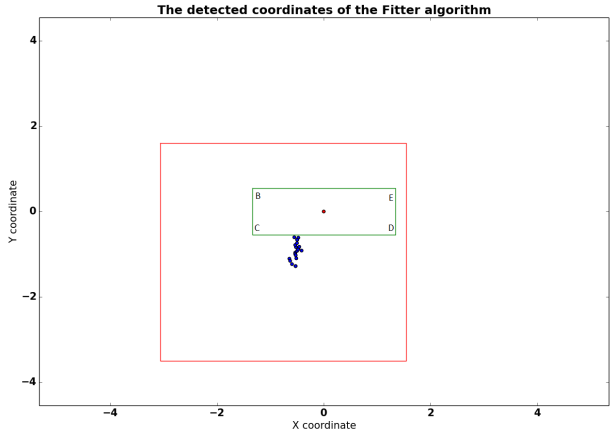


(a) The states and coordinates detected by the Fitter algorithm for the HTC mobile device in experiment 4(before)

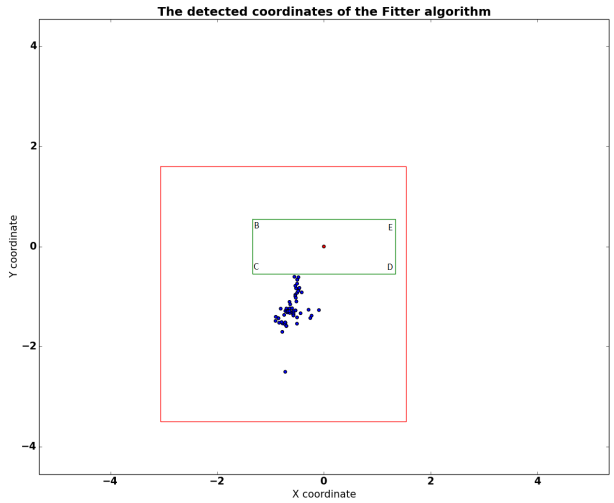


(b) The states and coordinates detected by the Fitter algorithm for the HTC mobile device in experiment 4(after)

Figure 5.7: The states and coordinates detected by the Fitter algorithm for the HTC mobile device in experiment 4(before and after)



(a) The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 4(before)



(b) The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 4(after)

Figure 5.8: The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 4(before and after)

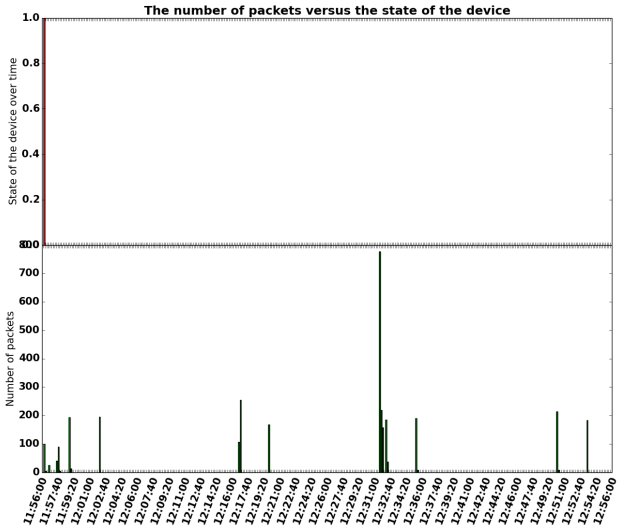
5.5.2 iPhone

Figure 5.9[2] presents the comparison between the performance of the points before and after the algorithm was fixed. As mentioned before, the algorithm initially computed only one point for this experiment, which could

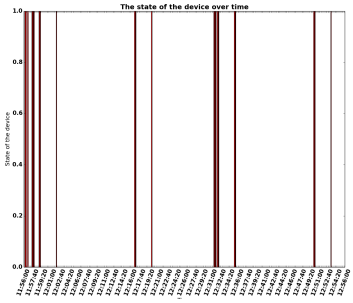
[2]This figure was separated in two plots, because the matplotlib package for python sometimes shrinks the image and the time intervals with detections do not overlap with the ones with detections

be seen in the first time interval of 10 seconds with data packets. This seemed a poor result given the existence of sufficient data packets for the rest of the 21 time intervals. However, the new results seemed encouraging as one could draw the conclusion that if there are more than 4 available data packets in a time interval, then the algorithm may compute a data point.

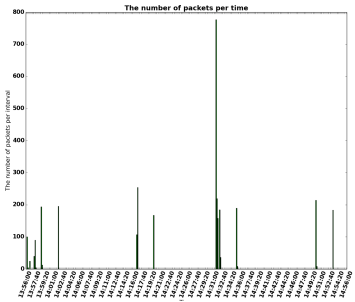
From a coordinate perspective, Figure 5.10 shows that a cluster of points outside the area of the table was added to the point found inside it after fixing the algorithm. This cluster is positioned on the right side of the reference point in the vicinity of the Delta router. This indicates as previously seen that even though the number of points increased, this did not mean that the performance improved as well. According to the results of experiment 4 for the iPhone case, the highest signal strength could be found in the case of Delta followed by Charlie which may have influenced the position of the computed data points.



(a) The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 4(before)

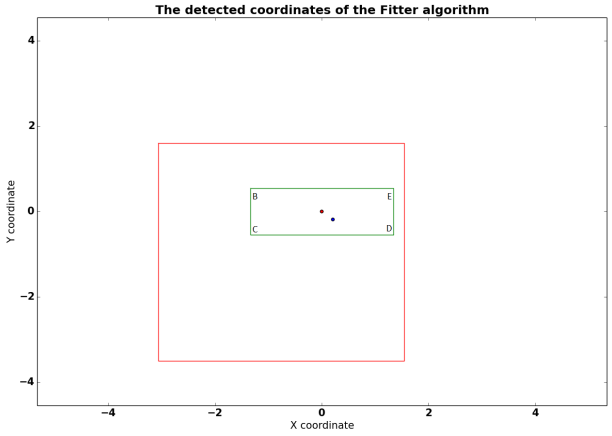


(b) The states detected by the Fitter algorithm for the iPhone mobile device in experiment4(after)

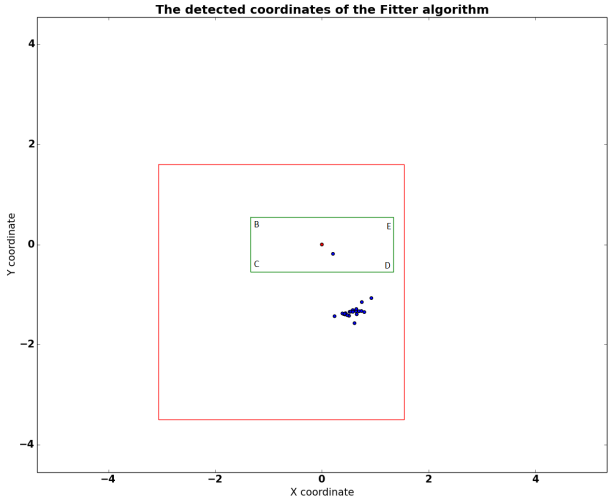


(c) The packets for the iPhone mobile device in experiment4(after)

Figure 5.9: The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 4



(a) The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 4(before)



(b) The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 4(after)

Figure 5.10: The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 4(before and after)

5.6 Conclusions

From the results of the experiments before and after the bug fix of the Fitter algorithm, the following conclusions can be drawn:

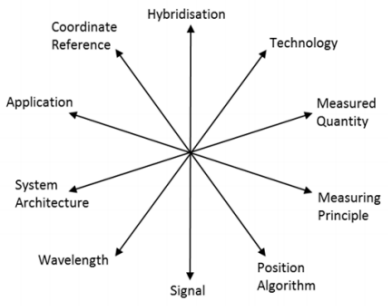
- The number of points reconstructed by the Fitter algorithm has increased significantly after the bug fix(e.g, a higher reconstruction efficiency)
- The accuracy of the average x and y coordinates is changed, but it was not improved in the majority of the cases
- The efficiency of the WiFi system and the efficiency of calculating coordinates have increased significantly
- There is a high probability that the Fitter algorithm is able to reconstruct the coordinates of the mobile devices if there are more than 4 data packets from 4 different drones
- Depending on the experiment and on the test, the extra reconstructed points by the Fitter are either points in the vicinity of the reference point or outliers
- WiFi tracking may be challenging in idle states of mobile devices
- Strategies need be developed for increasing the communication between mobile devices and drones

Chapter 6

Features of the system and comparison with other projects

In this chapter, the KPMG LAS system is compared with projects found in the literature by following the framework of Mautz[34] which contains the user and technical requirements an indoor positioning application should have. Crucial elements need to be taken into consideration according to Mautz[34] when actually designing and implementing an indoor positioning application. These requirements are shown in Figure 6.1a and Figure 6.1b. These figures show the multi-dimensionality of the optimization problem that the companies are confronted with.

(a) User requirements for indoor positioning systems(Mautz[34])



(b) Technical parameters for indoor positioning systems(Mautz[34])

6.1 Business-user requirements

Mautz[34] describes briefly each user requirement as follows in Figure 6.1a:

- **accuracy** (mm, cm, dm, meter, decameter level)
- **coverage area/limitations** to certain environments (single room, building, city, global)
- cost (unique system set-up costs, per user device costs, per room costs, maintenance costs)
- required infrastructure (none, markers, passive tags, active beacons, pre-existing or dedicated, local or global)
- **market maturity** (concept, development, product)
- **output data** (2D or 3D coordinates, relative, absolute or symbolic position, dynamic parameters such as speed, heading, uncertainty, variances)
- **privacy** (active or passive devices, mobile or server based computation)
- **update rate** (on-event, on request or periodically e.g., 100Hz or once a week)
- interface (man-machine interfaces such as text based, graphical display, audio voice and electrical interfaces such as RS-232, USB, fiber channels or wireless communications)
- system integrity (operability according technical specification, alarm in case of malfunction)
- robustness (physical damage, theft, jamming, unauthorized access)
- **availability** (likelihood and maximum duration of outages)
- scalability (not scalable, scalable with area-proportional node deployment, scalable with accuracy loss)
- **number of users** (single user e.g., total station, unlimited users e.g., passive mobile sensors)
- intrusiveness/user acceptance (disturbing, imperceptible)
- approval (legal system operation, certification of authorities)

Several user requirements are discussed here in more detail with respect to the ones presented by Mautz. These requirements are the following: accuracy, coverage, market maturity, integrity, availability, update rate, system latency, interface, data output, privacy, costs, and number of users.

Accuracy

The accuracy requirement represents the most important measure, if it is viewed from a user perspective. In order to compare system performances, a useful metric for the quality of positions is the computation of the standard deviation(RMSD-*Root Mean Squared Deviation* or RMSE-*Root Mean Squared Error*):

$$\sigma_P = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{P}_i - P_i)^2}$$

,

where n is the number of estimated position vectors $\hat{P}_i$ and P_i the position vector predicted by a modal of the localized node i, or, if only one single location is estimated, P_i is replaced with a single position vector P_0.

Low accuracy can be defined as the standard deviation $\sigma_P > 10m$ and *high accuracy* as $\sigma_P < 1cm$. Even though, the accuracy represents a key driver for most applications. It should be viewed in the context with the other performance parameters.

The main metric used for evaluating the **accuracy** of the KPMG WiFi tracking system is the root mean squared error($RMSE$) like the metric proposed by Mautz. According to the results of the experiments and the CHEP data, the $RMSE$ differs by experiment and test. Due to the fact that the coordinates are calculated by means of the Fitter algorithm in Chapter 3, Chapter 4, and Chapter 5, the main factor that affects the quality of the results represents the recorded signal strength by the drones, which is influenced itself by several factors. Other factors that also play a key role are the following: device type, device state, position of the mobile device, shape of the area under analysis, and the configuration of the drones, etc.

The results obtained from the CHEP conference showed that the accuracy for a lying still on a table iPad device was 7.8m for the x coordinate and 6.4m for the y coordinate. However, the results obtained from the experiments revealed a $RMSE$ smaller than 1m. Nevertheless, these experiments where performed with the drones close to the mobile(WiFi) devices. It might be that the mobile(WiFi) devices are detected with a higher $RMSE$ as long as it is not close to the drones.

It may be the case that the results obtained for the CHEP data have larger RMSE values of the x and y coordinates. This assumption is made based on the validation experiments which revealed that after the bug fix, the $RMSE$ increased. Verifying this assumption was not possible, because a new dataset of the CHEP could not be obtained after the bug fix and, thus, the models could not be applied.

The KPMG LAS system is compared with similar projects from the literature and public information of the main competitors. From a literature perspective, Bahl and Padmanabhan's system RADAR[20] has an accuracy of $2-3$m. From a competitors' perspective, an indoor positioning system developed by Polestar company represents NAO Campus, which promises an accuracy that ranges from 2.1m up to 5m[27]. The Infsoft company promises on their website[32] an accuracy of 1m. [1]

Coverage

The coverage describes the spatial extension where system performance must be guaranteed by a positioning system. One of these categories should be specified:

[1]It may be the case that the public information of the main competitors is not reliable for comparison due to marketing strategies.

- **Local coverage** - small, well-defined, limited area which is not extendable
- **Scalable coverage** - systems with the ability to increase the area by adding hardware(deployment of additional sensors)
- **Global coverage** - system performance worldwide or within the desired/specified area. Only the GNSS systems and celestial navigation belong in this category

The **coverage area** is dependent mainly on the detected signal strength of the mobile devices and the environmental limitations within this area. The environmental limitations are the elements such as windows, furniture, and walls that may interfere with the way the signal strength is recorded. Our experiments showed that sometimes these obstacles could increase or decrease the magnitude of the signal strength.

According to Subsection 4.2.2, more than 30 meters distance was enough to detect a mobile device. Nevertheless, the 30 meters were measured on a hallway where there were no obstacles. This value may change if these environmental limitations stand in the way. When comparing with the literature, Nuaimi and Kamel[12] argue that the range of existing systems go from 5 to 50 meters, as providing a system with a coverage of more than 60 is very challenging.

The category in which the KPMG WiFi tracking system belongs to is the scalable coverage systems, the systems with the ability o increase the area under analysis by adding drones. If we assume that the area under analysis has a square shape(the distance between the drone and the WiFi device is equal to 30m and half of the diagonal of the room) and based on the results of the experiments, then the coverage area may be approximately $1800m^2$.

Market maturity

The KPMG WiFi tracking system has been developed for less than 6 months from a **market maturity** perspective. It represents a product in development as it is tested only in small experiments like the ones developed within this work. Even though the performance seems to be promising, this products needs more testing in a real-life environment. For example, the system will be tested at the KPMG canteen.

Integrity

Integrity relates to the confidence which can be placed in the output of a system. Integrity risk is the probability that a malfunction in the system leads to an estimated position that differs from the required position by more than an acceptable amount(the alarm limit) and that the user is not informed within the specified period of time(time-to-alarm).

According to Mautz[34], regulator bodies have studied and defined integrity performance parameters in some sectors such as civil aviation, however, in other sectors, including those relating to indoor navigation it is more difficult to find quantified integrity parameters.

Availability

Availability is the percentage of time during which the positioning service is available for use with the required accuracy and integrity. This may be limited by random factors (failures, communications congestion) as well as by scheduled factors(routine maintenance). Generally, one of the following three levels could be specified, although this will depend on the particular application:

1. low availability: $< 95\%$
2. regular availability: $> 99\%$
3. high availability: $> 99.9\%$

To achieve availability, it is assumed that continuity, accuracy, and integrity requirements are fulfilled. Application descriptions usually include specification of availability, whereas system developers usually do not specify an availability figure.

As mentioned before, the KPMG LAS represents an experimental project which has not yet extensively been tested in a real-life environment. This means that assessing the *availability* of the system is difficult. However, it can be affirmed that the entire system was tested at the CHEP conference for 4 days in row and it was affected by random factors such as failures and congestion problems leading to a low availability. Due to the fact, that this project represents a pilot, the desire is to increase the availability to $> 99.9\%$.

Update rate

The update rate is the frequency with which the positions are calculated on the device or at an external processing facility. The following types of measurements rates exist:

1. **periodic**: regular update, specified in an interval (unit e.g., (Hz))
2. **on request**: triggered by the user or by a remote device.
3. **on event**: measurement update initiated by the local device when a specific event occurs, e.g., when a temperature sensor exceeds a critical threshold.

The KPMG WiFi system has an **update rate** of 10 seconds in which the positions are calculated. From a measurement rate type, this system enters in the periodic category as it synchronizes with the way the algorithms work, which, as mentioned before, use a buffer of 10 seconds time interval to compute the coordinates of a mobile device. From the rate of update perspective, it would be good in the future if the 10 second time interval is lowered to a smaller time interval. It may the case that the accuracy of the coordinates may be improved and the rate of update will be faster.

The update rate may suffer modifications depending on the tracking type, "static" or in "motion". On one hand, if the tracking type is static(the device does not move), more data packets may be needed to increase the

accuracy of the detections. On the other hand, if the tracking type is in motion(device is moving), then there are less data packets per unit time, which may decrease the accuracy of the coordinates.

System latency

The system latency describes the delay with which the requested information is available to the user. The latency can have the following values:

- real time: Does not tolerate "perceivable" delays. It is the most demanding latency requirement. It is necessary for navigation and almost all indoor positioning applications.
- sooner the better: Requires the system's best effort.
- sooner the better with an Upper Limit: Requires the system's best effort but the system must be designed to limit the maximum delay to a specified threshold.
- post processing: No specific time of delivery is defined.

When looking at the **system latency** perspective, the information related to what the main areas of interest are. The heatmap uses the reconstructed coordinates of the algorithm. Thus, the delay represents only from the amount of time it takes to the algorithm in order to compute the coordinates of the mobile device. The KPMG system may enter the post-processing category.

Interface

The KPMG LAS platform contains a user-friendly **web interface/dashboard** which consists of several parts. The heatmap plays a key role. It has on the background the floor plan of the area of coverage. On this floor plan, the areas where the mobile devices are detected are plotted. The heatmap shows this activity on an aggregated level and not on an individual device level. This way the privacy element is handled properly as the intention is only to count the number of devices in the areas of interest. These areas of interest enable the clients identify whether the current display layout is attractive enough for the clients, what are the most interesting and visualized products by the visitors, and where the clients spend more time e.g., at the cash-register, in a queue, etc.

The KPMG LAS dashboard also provides statistics with respect to the number of visitors in the area, as well as an intuitive plot which shows how this number varies in time. As well as, the percentages of the clients split by areas of interests are shown by means of a pie chart plot. In addition to this, the box-plot indicates the dwell time of the visitors and the results are presented on aggregated level.

Data output

The **output data** of the KPMG LAS platform contains $2D$ data with only x and y coordinats that are visualized on an aggregated level in the

heatmap, as well as the timestamp, the errors for the coordinates of the algorithm. It is foreseen that the z coordinates is also calculated by the algorithms. At the moment, the z coordinate is equal to 0, thus it might affect the accuracy of the computed coordinates. In addition to this, not all the mobile devices are located on the same axis.

Privacy

KPMG has put a lot of effort in ensuring that the data privacy is handled carefully. The KPMG LAS platform is designed to comply with the Dutch Data Protection Directive and contains a number of measures to ensure that it is virtually impossible to identify a single individual. In addition to this, the KPMG LAS platform provides opt-out(in) capabilities for those individuals who do not wish to be tracked. The system filters out the opt-out devices and retains only the ones that are opt-in. A randomiser is also forseen to be implemented which adds a random perturbation/statistical noise to the results of the analyses making the identification process virtually impossible(Aircloak).

Number of users

The web-interface of the KPMG LAS platform can be accessed by multiple *users* at the same time. However, the maximum number of users has not been assessed yet. Each user requires a username and a password in order to log in to the system.

6.2 Technical parameters

Besides, the user requirements which refer mainly to the companies interested in purchasing the IPA(Indoor positioning applications), there are still some technical parameters that need to be taken into consideration. Mautz described these as the following:

- level of hybridization(single modality, two different sensors, highly hybrid sensor fusion).
- technology(optical, inertial, magnetic, sound etc.)
- measured quantity(direction, distance, signal amplitude, acceleration, time)
- basic measuring principle ((tri)lateration, (tri)angulation, fingerprinting, cell of origin, dead-reckoning)
- positioning algorithm used (multidimensional scaling, multilateration, heuristics)
- signal used (sound waves, electromagnetic waves, magnetic field strength)
- signal wavelength (visible light, infrared, radio frequencies)
- system architecture (central or distributed systems)
- application (navigation, surveying, industry tracking, metrology)

– coordinate reference (local, global, object or sensor coordinate system)

According to the same author[34], the values of the performance parameters are sometimes not determinable, because they depend on various factors, circumstances, and conditions. The definitions of the positioning requirements parameters according to the same thesis[34] are presented, in order to have a reference point of how these systems should typically behave.

From a technical requirements' perspective, the following requirements perspective the following will be taken into consideration: technology, basic measuring principle, application, and signal strength.

Technology

The technology for the KPMG LAS is based on the detection of the WiFi signals, data that are recorded by the WiFi drones and that are sent to the KPMG Analytics Visualization and Environment Platform(KAVE), where the data is stored, validated, and calibrated. The output data obtained from the algorithms is visualized on a dashboard in a web-interface.

Basic measuring principle

The main measuring principle is trilateration, which represents a technique for identifying the location of a device if the distances of at least three references are known. The Fitter algorithm combines the trilateration technique with nonlinear Chi-square fit and Newton-Rapson technique to estimate the best possible parameters for the non-linear relationship between the signal strength and the distance[55].

Application

There are multiple applications of the KPMG LAS to the business side. Firstly, the interest areas are revealed based on which customer paths can be identified. Both enhance the understanding of the customer base. In addition to this, a new store layout can be optimized. For example, the visitors/customers may find in their way products that are sold in associations with the most purchased ones or they may be "required" to follow a different path in their way to certain products. These products can be determined based on the market basket analysis technique. Such techniques offer insight in the relationships between products that are purchased together. This insight combined with the information extracted from the WiFi tracking system can be used for customer profiling and targeted advertising.

Due to privacy restrictions, the tracking system will not be evaluated on an individual device level to control to which extend the individual data is used. However, besides the opt-out feature, KPMG also supports an opt-in feature for clients who would like to be contacted after since they

are interested in a certain product and they would like to have much more information.

6.3 Conclusion regarding the system

The performance of the KPMG LAS platform was evaluated from both business and technical perspectives. This system is currently a pilot which seems to have a lot of potential. From a business perspective, this system brings insight into the interest areas of the customers, an estimation of the number of detected devices, and several other statistics relevant for matching product offers to customers.

6.4 Implications of Apple's decision to implement random Mac address on iOS8

At the Apple Worldwide Developers Conference in June 2014, David Stites and Katie Skinner announced the changes of the WiFi scanning behavior in the iOS8. It will will use random, locally administrated MAC addresses for probe requests(sub-type $0x4$) and responses(sub-type $0x5$). According to Apple's product security and privacy representatives' presentation[47], the MAC address used for the WiFi scans may not always be the devices' real(universal) address.

In this section, the potential implications of these changes are evaluated. Greg Sterling from Marketingland[53] believes that these decisions are mainly targeted such that Ad networks, WiFi hotspots, and other third parties will not be able anymore to have access to the real MAC Iphone iOS 8 and to combine them with other datasets. Thus, individuals cannot be associated anymore with particular locations for later re-targeting. It may also be the case that Apple tries to get people's consent that their MAC addresses can be used.

Another potential implication of this change may affect the statistics related to the dwell time of the Iphone mobile devices that are not connected to the network. However, this will not have implications on the counting process, if the devices are connected to the WiFi network. In this case, the MAC address is not random anymore and all the statistics can be calculated based on that. In addition to this, the Android mobile devices are not affected by this decision. The main competitors in the indoor/outdoor positioning industry surprisingly support this decision even though this may influence the various analyses that are performed.

It may be the case that Apple tries to increase their market share by attracting potential clients that are concerned with privacy. However, the company has bought in 2013 the WifiSlam start-up. They built for indoor positioning technologies that pinpoint the location of the smartphone(and the location of your friends) in real-time with an accuracy of 2.5m by using only ambient WiFi signals that are already present in buildings. Besides this, they developed location-based mobile apps that could engage with

users at the scale that personal interaction actually took place. Other applications ranged from step-by-step indoor navigation, to product-level retail customer engagement, to proximity-based social networking.

Chapter 7

Contributions, limitations, conclusions, recommendations

7.1 Contribution to the literature

This work analyzes the performance of a new WiFi tracking system for location aware services from both business and technical perspectives. In the literature, most projects for location aware services are small pilot projects that do not involve large amounts of data. Instead, in this research data were analyzed from both real-life environment(CHEP conference), but also controlled experiments which zoomed in and described how mobile devices communicate with the drones in different situations. The most important factors that influence the way the coordinates are calculated in controlled environments were identified based on the literature and on the controlled experiments.

This research also proposes a strategy for analyzing the performance of the WiFi tracking systems by looking at the most important statistics both on individual and aggregated levels, which could not be found in the literature. It evaluates into depth the performance of the algorithms and proposes models for calculating the probability of detecting mobile devices. In addition to this, a visualization over time, statistics, and indicators are created to assess the performance of the WiFi tracking system and the algorithms.

From a business perspective, this study analyzes the WiFi tracking system based on the framework proposed by Mautz[34] for indoor positioning algorithms. In addition to this, it contains information related to the main competitors on the indoor tracking industry, but also projects from the literature.

7.2 Limitations

This research has several limitations which are explained chronologically.

One of the main limitations within Chapter 2 was that the CHEP datasets used for analysis lacked relevant data which could be used for the analysis. Therefore, the accuracy was evaluated only based on one device(an iPad device). In addition to this, the actual coordinates of the iPad device may not be correctly calculated. This may have an impact on the bias of the average coordinates for both x and y coordinates. For a better understanding of how this WiFi tracking system works, the entire tracking system should be tested in a real-life environment and on several more devices which have their states and positions known.

In Chapter 3, two models were proposed for computing the probability of detecting mobile devices. The probability of detecting the mobile devices was calculated on a 5 minute level, but also on a 10, 30, and 60 second level. The results of the models were obtained based on the datasets from the old version of the Fitter algorithm which had a bug. Due to this bug, the algorithm did not compute all the points even though there were sufficient available data packets. It may be the case that the results of the models are obsolete and that the probability as well as the indicatiors may increase after the bug fix. Unfortunately, due to computer and time limitations, the new Fitter algorithm after the bug fix could not be applied on a day level for the CHEP data. The Fitter algorithm was run more than 6 times on the raw dataset from the CHEP conference. However, it failed to compute the entire dataset. Due to the fact that the history obtained based on the CHEP data is not relevant to KPMG, redoing the analysis would not bring any insight to this research.

Chapter 4 contained the design and implementation of the experiments, which can be structured in two main categories: tracking devices and testing the relationship between signal strength and distance.

The tracking experiments had the same configuration across the entire tests. Nevertheless, there was a difference between the experiment 1 and experiments 3 and 4, respectively, from the perspective of used drones. For experiment 1, the Foxtrot drone was used which, according to the results, it behaved differently compared to the other drones. A possible reason for this behavior may be that Foxtrot had technical problems. Thus, for experiments 3 and 4, it was replaced with the Beta drone.

There are several other limitations related to the design and configuration of the experiments. All three experiments were performed in an isolated room, where four different drones were placed on the table. The reason why four different drones were used was that the Fitter algorithm requires four different drones in order to compute a data point for a particular mobile device. The position of the drones was higher than the one of the mobile device. The mobile device was placed on a wooden table. It may the case that the signal strength may have been attenuated by this wooden table. This table was also close to the windows of the room which may also have had an influence on the magnitude of the signal strength. Another limitation of these experiments constitutes the fact that only one position

of the cell phone was tested, laid on the table. It may be the case that the signal strength may have a different behavior with other positions.

Experiment 2 also faced several limitations. Firstly, the shape of the hallway where the drones were placed may have influenced the magnitude of the signal strength. Secondly, the hallway represents a public space in the office. Thus, it may be the case that the people's bodies interfered in the recording process. Lastly, this experiment was performed with only two drones, compared to the other three experiments.

7.3 Conclusions

The KPMG WiFi tracking system works. The *Fitter* and *Trilaterator* algorithms seem to have similar performance. However, the *Fitter* algorithm performs better from a resolution perspective and, thus, the rest of the analyses and models were created based on its results.

Two different models were developed for calculating the probability of detecting a mobile device: the "basic" model and the "cut-off" model. With the first model, the probability of detecting a WiFi device seems to be underestimated, while the results of the second model show that the probability of detecting a WiFi device increases considerably, when larger time intervals are used for detecting devices.

Four experiments were developed for analyzing how the drones listened passively to WiFi devices. It was found that the signal strengths of the devices were measured differently by the drones when placed in different positions. In addition to this, no pattern could be found with respect to the recorded signal strengths and different positions. It could be seen that the mobile devices sent less packets when they were in an idle state compared to when an application was used or when the phone was active, but not connected to a network. The experiments also revealed that the Fitter algorithm did not compute data points sometimes even though there were sufficient data packets and that a bug was in the algorithm.

The comparison between the results before and after the bug fix revealed that the algorithm did not reconstruct some of the data points even though there were sufficient data packets and it seemed to perform sometimes as a quality cut for several points.

Based on the results of the experiments, the bug fix in the algorithm impacts the outcome of the models for calculating the probability of detecting mobile devices in several ways. On one hand, the number of reconstructed data points that will be computed may increase considerably and, thus, the delta time between consecutive detections may decrease and the distribution of the delta time may be different. In addition to this, it may the case that the threshold of 5 minutes is not appropriate anymore, given the fact that the mobile devices are detected almost every 10 seconds when there are sufficient packets detected by the drones. Besides the delta time, the number of mobile devices detected every 10 seconds also may increase, because additional points are computed for a mobile device. This impacts also the shape of the plot of the total detected devices, which seemed to

have a time dependent shape with several peaks, as well as the arrivals, and departures.

The $\Delta time$(the time between two consecutive detections of a device) could not be entirely explained as proposed in the beginning. There are several explanations that can be given to it depending on the type of used algorithm. If the Fitter algorithm was used before the bug fix, it may be the case that a large $\Delta time$ can be explained by the bug. However, it may also be the case that the mobile device can be in an idle state, switched off, or not detected.

In principle, assuming that the Fitter algorithm after the bug fix is used and the information extracted from the experiments, it is not possible to explain what is happening within a large $\Delta time$, whether it is dwell time, walking time, missing data, or any other combination between them. What it can be affirmed is that, after the bug fix, the Fitter algorithm is able to reconstruct coordinates if there are more than 4 data packets recorded from the drones with a certitude of more than 90%.

In the end, the performance of the KPMG LAS platform was evaluated from both business and technical perspectives. This system is a pilot which seems to have a lot of potential given the time within it was developed. From a business perspective, this system brings insight into KPI's such as the interest areas of the customers, an estimation of the number of detected devices, and dwell time of the visitors. However, there are several improvements that need to be made mainly with respect to the resolution of the calculated coordinates in order to increase the accuracy of the reconstructed data points.

7.4 Recommendations for improving the performance of the WiFi tracking system

Based on the research performed so far on the KPMG LAS system, certain recommendations are made for improving the performance of the system which will be briefly explained:

1. Calibration of the drones to equalize the received signal strength across the drones, because the signal strength represents a key factor in reconstructing the coordinates of mobile devices
2. More experiments should be performed in order assess the performance of the system such as:
   (a) WiFi devices in different states such as idle, active, or not connected to the network with the owner walking at different speeds
   (b) WiFi devices held in different places such as hand, pocket, and bags
3. Investigating more expensive hardware to ensure the drones are operating in a reliable manner
4. Thoroughly testing the Fitter algorithm for bugs and errors with different settings in order to verify the consistency and accuracy of the data

5. Improving the way the Trilaturator algorithm converts the signal strengths into distances(fit the relationship between signals and distances)
6. Generating documentation of the entire WiFi tracking system
7. Developing mobile applications which clients can use for indoor positioning of their interest areas that may stimulate the communication between mobile devices and drones to improve tracking
8. Building simulations of how the signal strengths can be measured in different environments, because the environment may have a big impact on the recorded signal strength
9. Testing the number of counted devices by the system from the web-interface to ensure the accuracy of the detections
10. Developing a system that can automatically calculate the efficiency and potential failures of the entire system in order to acknowledge potential problems
11. Building algorithms and simulations that calculate the dwell time to improve in order to prevent underestimation or overestimation
12. Integrating the WiFi tracking with other technologies such as GPS, Bluetooth, movement sensors, and cameras to improve the quality of the tracking

Appendix A

Appendix

A.1 Terminology

In this section, we briefly describe the most important parameters based on which we will compute different statistics.

1. The $time_{start}$ represents the start time of the period that we will analyze the system, while the $time_{stop}$ represents the end time of the period we will analyze the system.
2. The interval of [$time_{start}$, $time_{stop}$] can be either a day, half a day, or any other desired interval time. Usually, we define this interval on a 5 minute level, but it can also be expressed on a second level.

   The starting and ending time have the following structure: "'yyyy-mm-dd HH:MM:SS"', where y stands for year, m for month, d is for day, H for hour, M for minute, S for second.
3. The *time_interval* represents the time moment for which we zoom-into the selected period ([$time_{start}$, $time_{stop}$]). It can be expressed both in minutes and seconds.
4. A *detected device* represents a device which is recorded by the LAS. Thus, records related to this device can be found in the database.
5. An *arrival* represents the first time moment when a mac address is seen for the first time in the database within a selected period of time. This means that before this detection there was no previous information related to this mac address.
6. A *departure* is the last time moment or a moment after which a mac address will not be detected anymore. Hence, there will be no other record in the database besides this one and we say that "'it has left the system"'. This means that this mac address will never be seen in the selected period.
7. A *missing* mac address represents a mac address which had an arrival, but which did not have a departure. However, this mac address is not being detected within certain time intervals. This situation can occur, due to several reasons like: the drones do not function, the

drones are busy, the mac address left the area of detection, or the device is idle.

8. The *ideal number* of mac addresses represents the number of mac addressed that should ideally be detected at a certain time moment given the known dataset.
9. The *intervals* represents the number of intervals of user selected time interval($time\_interval$) that are obtained within a time period([$time_{start}$, $time_{stop}$]):

$$intervals = \frac{(time_{start} - time_{stop})}{(60 \times time\_interval)}$$

We divide by 60 in order to transform the result from seconds to minutes. The same formula is used but without dividing by 60 if we want to use seconds instead.

10. The $Arrivals_i$ constitutes the number of mac addresses that are seen first within the time interval $i = [t, t + time\_interval]$.
11. The $Departures_i$ is the number of mac addresses which are detected for the last time within $i \in [t, t + time\_interval]$.
12. The $Total\_detected_i$ is the equivalent of the total number of unique devices which are being detected within $i = [t, t + time\_interval]$.
13. The *matrix of detection* represents a matrix which contains the number of detections of a mac addresses per time interval. It has the following format: the rows represent the name of the unique hashed mac address, while on the columns we can find the time interval number, which ranges in [0, $intervals$]. Let m be the number of unique mac addresses that can be analyzed and n be equal to the number of intervals.

$$Matrix_{detections} =$$

$$\begin{array}{c} \\ Mac_0 \\ Mac_1 \\ \vdots \\ Mac_m \end{array} \begin{array}{c} \begin{array}{cccc} i_0 & i_1 & ... & i_n \end{array} \\ \begin{bmatrix} \#\_detections_{00} & \#\_detections_{01} & \dots & \#\_detections_{0n} \\ \#\_detections_{10} & \#\_detections_{11} & \dots & \#\_detections_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \#\_detections_{m0} & \#\_detections_{m1} & \dots & \#\_detections_{mn} \end{bmatrix} \end{array}$$

If the $\#\_dectections_{ij}$=0, where $i \in [0, m]$ and $j \in [0, n]$, and there was an arrival, but no departure, then this means that the Mac_i is not being detected anymore, thus it is missing.

14. The *behavior of a mac address* represents a compressed way to describe how a mac address has been or not detected, which contains information about its arrival and departure during a time interval of a time period, as well as the number of consecutive detected intervals followed by the number of consecutive missing intervals. This compressed way of describing a mac address is very efficient when calculating the statistics for very small time intervals as 10, 30, or 60 seconds, where there are a lot of time intervals.

A.2 Statistics

Let there be *time_interval* as the user-selected time interval and the time period $[time_{start}, time_{stop}]$ within which we will compute the following statistics:

1. **The number of unique arrivals per selected time interval**

   $$Arrivals_{[t,t+time\_interval]} = No.\ mac\ addresses\ with\ first\ appearances \in [t, t + time\_interval]$$

   , where $t \in [time_{start}, time_{stop}]$
2. **The number of unique departures**

   $$Departures_{[t,t+time\_interval]} = No.\ mac\ addresses\ with\ last\ appearances \in [t, t + time\_interval]$$

3. **The number of detected devices per selected time interval**

   $$Total\_detected_{[t,t+time\_interval]} = Arrivals_{[t,t+time\_interval]} + Departures_{[t,t+time\_interval]}$$
   $$+ Actual\_macs_{[t,t+time\_interval]} - Mac\_in\_out_{[t,t+time\_interval]}$$

   where $Mac\_in\_out_{[t,t+time\_interval]}$ represents the number of unique mac addresses that arrive and depart in the time interval [t,t+time_interval]. The reason why we need to correct the value with these mac addresses is that we add them twice: first for their appearance and second for their departure, however, it represents the same mac address.
4. **The number of devices that should have ideally been detected by the drones**

   $$Ideal_{[t,t+time\_interval]} = \sum_{i=[t_{start}, t_{start}+time\_interval]}^{[t-time\_interval, t]} (Arrivals_{[i]} - Departures_{[i]}) \tag{A.1}$$

5. **The number of missing mac addresses or the number of not detected mac addresses**

   $$Missing_{[t,t+time\_interval]} = Ideal_{[t,t+time\_interval]} - Total\_detected_{[t,t+time\_interval]}$$

   - $\text{Departures}_{[t,t+time\_interval]}$
6. **The probability of detecting a device per time interval**

   $$P\_detected\_devices_{[t,t+time\_interval]} = \frac{Actual_{[t,t+time\_interval]}}{Ideal_{[t,t+time\_interval]}} \tag{A.2}$$

, where

$$\text{Actual}_{[t,t+time\_interval]} = Total\_detected_{[t,t+time\_interval]} - Arrivals_{[t,t+time\_interval]} - Departures_{[t,t+time\_interval]} + Mac\_in\_out_{[t,t+time\_interval]} \quad \text{(A.3)}$$

and represent the mac addresses which arrived in a previous interval, but have not departed yet.

7. **The corrected percentage of detecting a device per time interval**

$$P\_detected\_devices_{[t,t+time\_interval]} = \frac{Actual_{[t,t+time\_interval]}}{Corrected\_Ideal_{[t,t+time\_interval]}} \quad \text{(A.4)}$$

where $Actual_{[t,t+time\_interval]}$ has the same formula previously mentioned, while the

$$Corrected\_Ideal_{[t,t+time\_interval]} = Ideal_{[t,t+time\_interval]} - Correction_{[t,t+time\_interval]} \quad \text{(A.5)}$$

, which represents a correction that we use in order to not underestimate the value of the percentage of devices detected.

8. **The average time difference between two consecutive detections**

This is computed in two steps. First, we calculate the average delta time per mac address:

$$Average\_delta\_time\_mac_i = \sum_{j=[time_{start},time_{start}+time\_interval]}^{[time_{stop}-time\_interval,time_{stop}]} \Delta time_{ij} \frac{}{\sum_{i=[time_{start},time_{start}+time\_interval]}^{[time_{stop}-time\_interval,time_{stop}]} \#\_detections_i - 1} \quad \text{(A.6)}$$

Then we calculate the average delta:

$$Average\_delta_i = \sum_{i=0}^{m} Average\_$$

$\Delta time\_\text{mac}_i \frac{}{m}$ (A.7) where m represents the number of unique mac addresses.

The average number of detected devices

$$\frac{\sum_{j=[time_{start},time_{start}+time\_interval]}^{[time_{stop}-time\_interval,time_{stop}]} \#\_detections_j}{n_j} \quad \text{(A.8)}$$

, where n represents the number of unique mac addresses detected for each j.

The pull distribution

The pull distribution has the following formula:

$$pull_x(i) = \frac{(x_i - actual_x(i))}{\sigma_x(i)}$$
$$pull_y(i) = \frac{(y_i - actual_y(i))}{\sigma_y(i)}$$

where the $x(i)$ and $y(i)$ represent the determined coordinates, the $actual_x(i)$ and $actual_y(i)$ are the actual measurements of a mac address, and the $\sigma_x(i)$ and $\sigma_y(i)$ are the errors of the algorithms.

A.3 Experiments

A.3.1 Experiment 1

A.3.1.1 Test 1

Time	Action	Other
14:55	Started the experiment	Direction towards Foxtrot and Delta
15:57	Pressed the home button	-
15:58	Pressed the home button	twice
14:59	Pressed the home button	-
15:01	Pressed the home button	-
15:02	Pressed the home button	-
15:04	Pressed the home button	-
15:05	Pressed the home button	the cell phone a bit shifted to the side
15:07	Pressed the home button	now it has the correct position
15:08	Pressed the home button	-
15:09	Pressed the home button	-
15:10	Pressed the home button	-
15:11	Pressed the home button	-
15:13	Pressed the home button	-
15:14	Pressed the home button	Stop the experiment

Table A.1: Experiment 1 Test 1

A.3.1.2 Test 2

Time	Action	Other
15:14	Started the experiment	Direction towards Echo and Delta
15:16	Pressed the home button	-
15:18	Pressed the home button	twice
15:19	Pressed the home button	-
15:21	Pressed the home button	-
15:22	Pressed the home button	-
15:23	Pressed the home button	-
15:26	Pressed the home button	twice
15:27	Pressed the home button	-
15:29	Pressed the home button	Stop the experiment

Table A.2: Experiment 1 Test 2

A.3.1.3 Test 3

Time	Action	Other
15:30	Started the experiment	Direction towards Echo and Charlie
15:32	Pressed the home button	-
15:33	Pressed the home button	-
15:34	Pressed the home button	-
15:36	Pressed the home button	-
15:37	Pressed the home button	-
15:38	Pressed the home button	-
15:39	Pressed the home button	-
15:40	Pressed the home button	-
15:42	Pressed the home button	-
15:44	Pressed the home button	Stop the experiment

Table A.3: Experiment 1 Test 3

A.3.1.4 Test 4

Time	Action	Other
15:45	Started the experiment	Direction towards Charlie and Foxtrot
15:46	Pressed the home button	Changed a bit its position
15:47	Pressed the home button	-
15:49	Pressed the home button	-
15:51	Pressed the home button	third times pressed
15:52	Pressed the home button	-
15:54	Pressed the home button	twice
15:55	Pressed the home button	Stop the experiment

Table A.4: Experiment 1 Test 4

A.3.2 Experiment 2

Test	$Time_{start}$	$Time_{end}$
$Test_1$	13:40	13:49
$Test_2$	13:50	14:00
$Test_3$	14:03	14:13

Table A.5: Experiment 2

A.3.3 Experiment 3

Test	HTC		Iphone	
	$Time_{start}$	$Time_{end}$	$Time_{start}$	$Time_{end}$
$Test_1$	11:40	11:56	15:00	15:16
$Test_2$	11:58	12:14	15:19	15:35
$Test_3$	12:20	12:36	15:40	15:56
$Test_4$	12:38	12:54	15:59	16:15

Table A.6: Experiment 2

A.4 Datasets

The used datasets depend on the experiment and the analyses that were performed. For simplicity, these datasets are explained by chapter as follows:

- Chapter 2, Comparison between the two developed algorithms
  - The Fitter dataset of the CHEP conference
  - The Trilaterator dataset of the CHEP conference
- Chapter 3, Modeling the time dependency of detecting WiFi devices
  - The Fitter dataset of the CHEP conference
- Chapter 4, Controlled table top experiments
  - Experiment 1
    - The dataset with the recorded signal strength for the iPhone mobile device
    - The Fitter datasets of the reconstructed points for this experiment before the bug fix
  - Experiment 2
    - The dataset with the recorded signal strength for the HTC mobile device
  - Experiment 3
    - The dataset with the recorded signal strength for the iPhone and HTC mobile devices
    - The Fitter datasets of the reconstructed points for the iPhone and HTC before the bug fix
  - Experiment 4
    - The dataset with the recorded signal strength for the iPhone and HTC mobile devices
    - The Fitter datasets of the reconstructed points for the iPhone and HTC before the bug fix
- Chapter 5, The performance of the Fitter algorithm
  - Experiment 1
    - The dataset with the recorded signal strength for the iPhone mobile device

    - The Fitter datasets of the reconstructed points for this experiment before the bug fix
    - The Fitter datasets of the reconstructed points for this experiment after the bug fix
- Experiment 3
    - The dataset with the recorded signal strength for the iPhone and HTC mobile devices
    - The Fitter datasets of the reconstructed points for this experiment before the bug fix
    - The Fitter datasets of the reconstructed points for this experiment after the bug fix
- Experiment 4
    - The dataset with the recorded signal strength for the iPhone and HTC mobile device
    - The Fitter datasets of the reconstructed points for this experiment before the bug fix
    - The Fitter datasets of the reconstructed points for this experiment after the bug fix

A.5 Experiments and tests

A.5.1 Experiment 1

A.5.1.1 Test 2

The second test is shown in Figure 4.1b. In this experiment, the cell-phone is on the table turned to the Echo and Delta drones. The start time of the Test 2 was at 15 : 13 until 15 : 29. The history of the actions that were performed during this test can be read in Table A.2.

As in the case of the previous test, we plotted the signal strength distribution over time and per drone. The results are visualized in FiguresA.1 and A.2. It can be observed that over time in this case the Foxtrot drone has the highest detected signal, followed by Delta, Echo, Charlie, and in the end Alpha, which is different from the previous case. In addition to this, Foxtrot has again a smaller number of records than the rest of the drones.

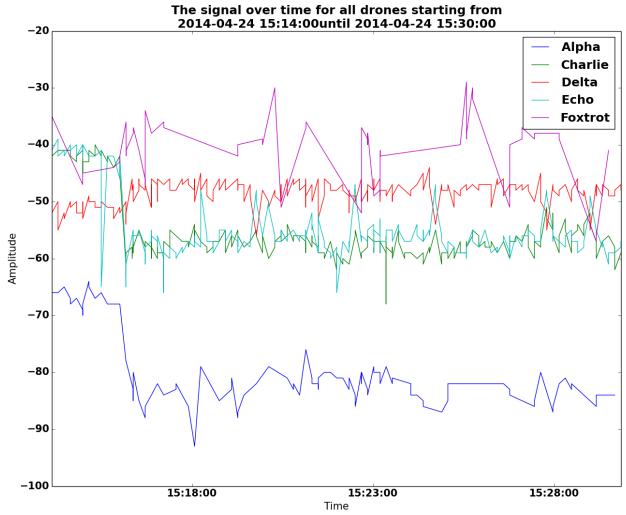


Figure A.1: The signal strength of the five drones during Test 2

From a signal strength distribution perspective, it can be observed that all the drones seem to have a Gaussian distribution shape. In contrast with Test 1, the highest average value of the signal strength can be found in the case of the Foxtrot drone, followed by the Delta drone with an average value of -48.6 ± 2.1, then by Echo -55.0 ± 5.2, and lastly Charlie -55.9 ± 5.2. These results may indicate that there is a difference between the signal strengths given the position of the cell-phone.

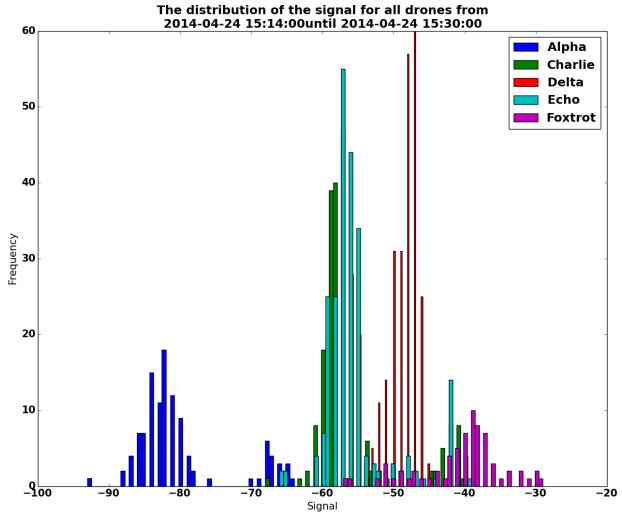


Figure A.2: The distribution of signal strength for each of the five drones during Test 2

By looking at the scatter plot in Figure A.3, it can be observed that most of the points are still on the table, where they form two clusters. It can observed that the cluster on the left resembles the one obtained in Test 1, with the x coordinates shifted to the left side and the y coordinates very close to 0. Besides this cluster, one can also see another one on the right side. These points are a bit shifted for both coordinates. An explanation for this result may be that the Fitter allocates different weights to the received signal strength by different drones depending on the magnitude of them. As seen in the previous plots, the Foxtrot drone seemed to detect the most powerful signal strength.

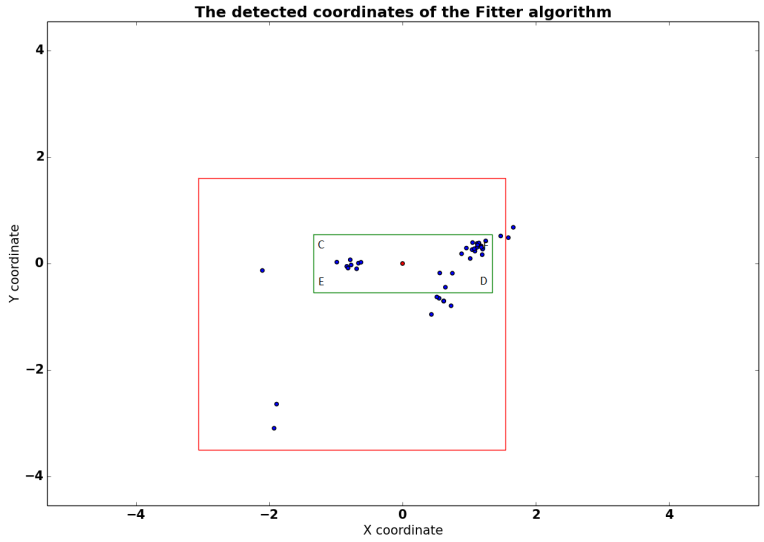


Figure A.3: The determined coordinates by the Fitter algorithm after Test 2

A.5.1.2 Test 3

The third test can visualized in Figure 4.1c. The cell-phone is on the table turned to the Echo and Charlie drones and the history of the actions that were performed in this test is in Table A.3. This third test started at 15 : 30 until 15 : 45. As it can be seen in Figure A.4, Foxtrot seems to detect again the highest signal strength and it seems to have a different shape than the others. By comparing the other drones with each other, Delta seems to have a higher signal than Echo and Charlie, followed in the end by Alpha. According to this plot, Echo and Charlie detected the same signal at the beginning of this test, but after a while they started differentiating themselves.

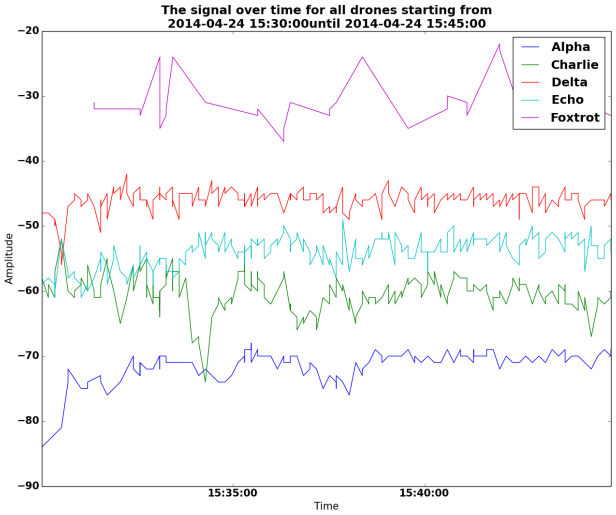


Figure A.4: The signal strength of the five drones during Test 3

On one hand, when looking at the distribution of the signal strengths in Figure A.5, the shape of Foxtrot's distribution does not resemble the Gaussian one, which may give us again an indication of the fact that this drone may have some technical problems. On the other hand, the other drones seem to have the shape of the normal distribution, with average values in this order: Delta, followed by Echo, Charlie, and in the end Alpha. Compared to the other tests, the Echo and Charlie distributions do not seem to resemble anymore and they are not centered anymore in a close average signal value. According to Table 4.3, the average value of the signal strength for Charlie was -60.3 ± 3.1 smaller than the one of Echo -54.2 ± 2.4.

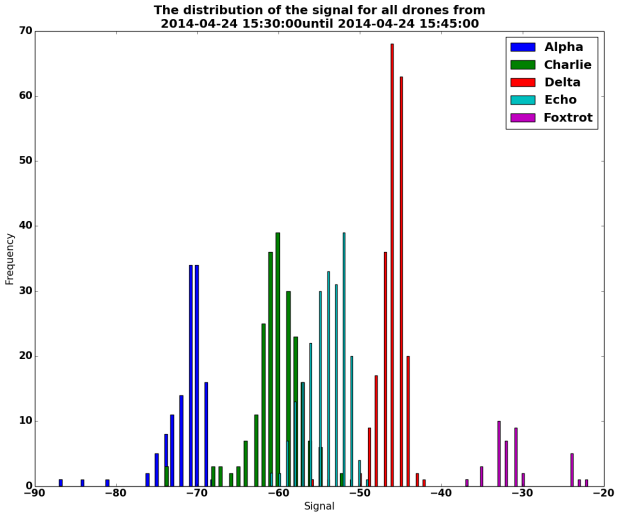


Figure A.5: The distribution of signal strength for each of the five drones during Test 3

The Fitter algorithm seems to perform less well for this test compared to Test 1 and Test 2, as only 2 points were calculated(Figure A.6). However, these points are detected in the area of the table. As it can be seen, one of the points seems to have the y coordinate close to 0, while the other one was detected in the area of the Foxtrot drone, which can be explained by the fact that the algorithm allocates a higher weight to the drone which detects the highest signal strength, which in this case is Foxtrot.

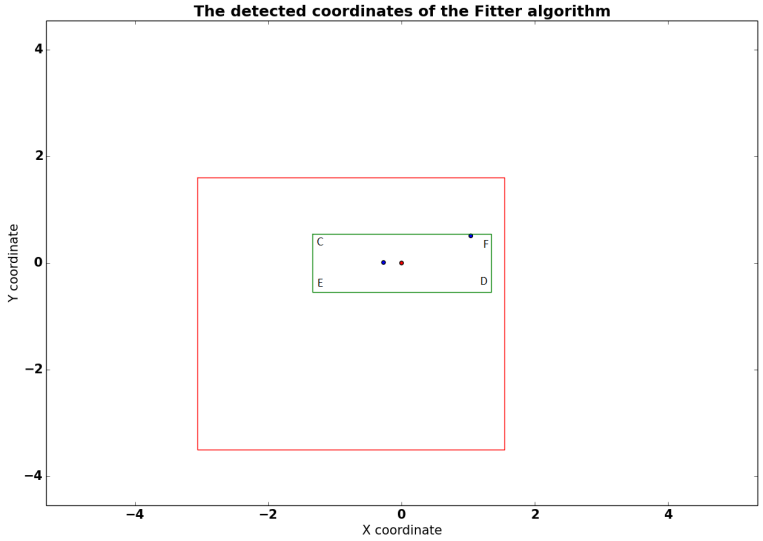
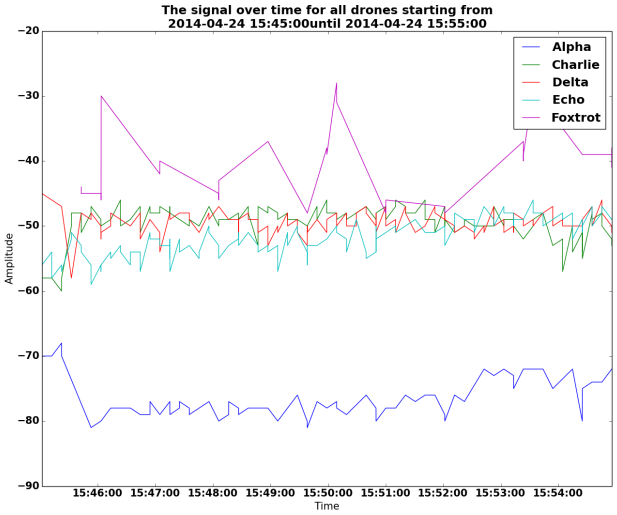


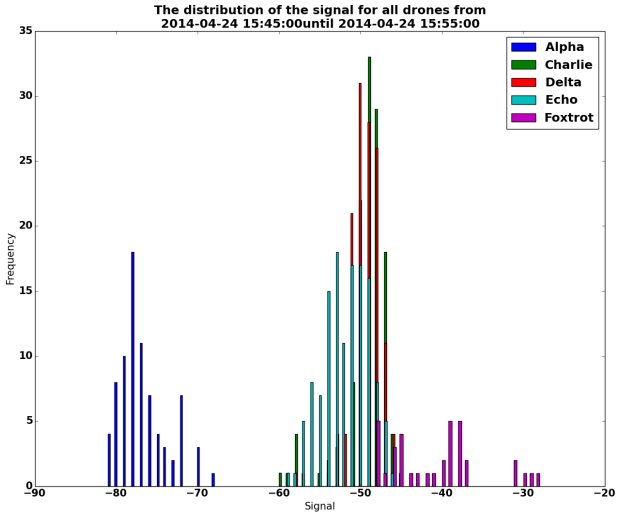
Figure A.6: The determined coordinates by the Fitter algorithm after Test 3

A.5.1.3 Test 4

The fourth test is visualized in Figure 4.1d. It started at 15:45 until 15:55. The cell-phone was placed on the table turned to the Charlie and Foxtrot drones and the history of the actions that were performed in this test is in Table A.4. According to Figure A.7a, the signal strengths of Charlie, Delta, Echo seem to intersect each other pretty much, especially in the case of Delta and Charlie. This seems to be an interesting aspect, as this result has not occurred before.



(a) The signal strength of the five drones during Test 4



(b) The distribution of signal strength for each of the five drones during Test 4

Figure A.7: The results of Test 4 experiment 1

When looking at the distributions of the signal strength, it appears that all the distributions except for Foxtrot seem to follow a Gaussian distribution. Charlie, Delta, and Echo seem to have these distributions centered in pretty similar average signal strength, which are -52.6 ± 6.0 for Charlie, -49.0 ± 2.1 for Delta, and -51.3 ± 4.3 for Echo(Table 4.3).

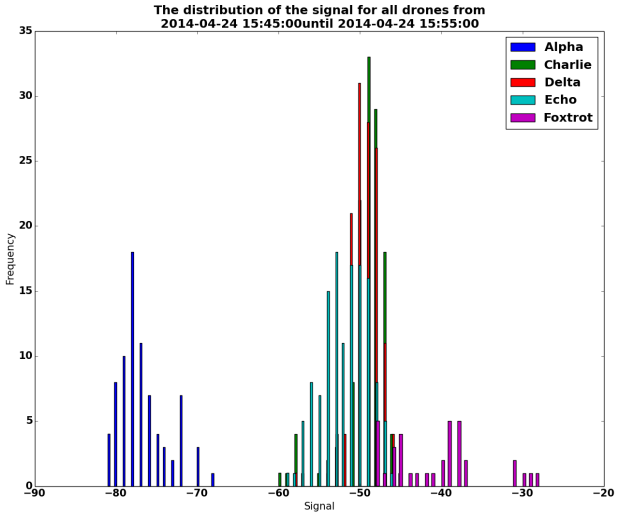


Figure A.8: The distribution of signal strength for each of the five drones during Test 4

The scatter plot of Test 4 in Figure A.9 shows that the Fitter algorithm detects the coordinates of the iPhone close to its actual point. Most of the resulting points form two different clusters on the table, which have the value of the x coordinate on average close to zero. However, there can be also seen points which are positioned outside the table, but still in the room.

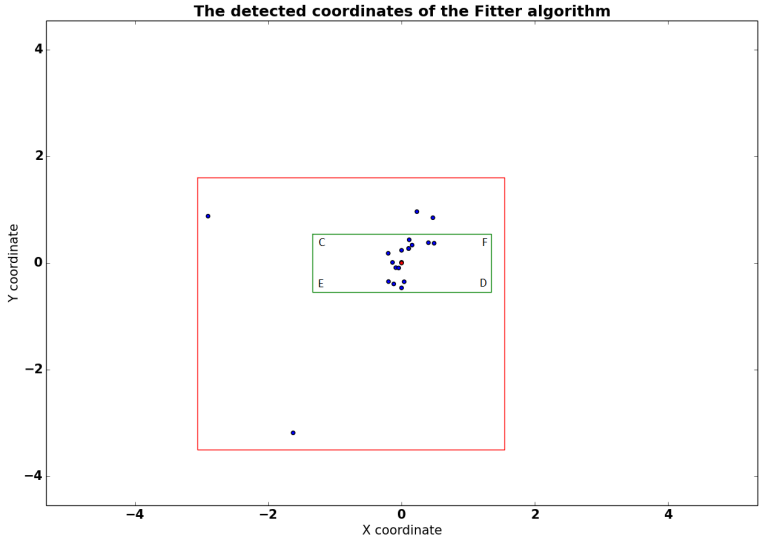
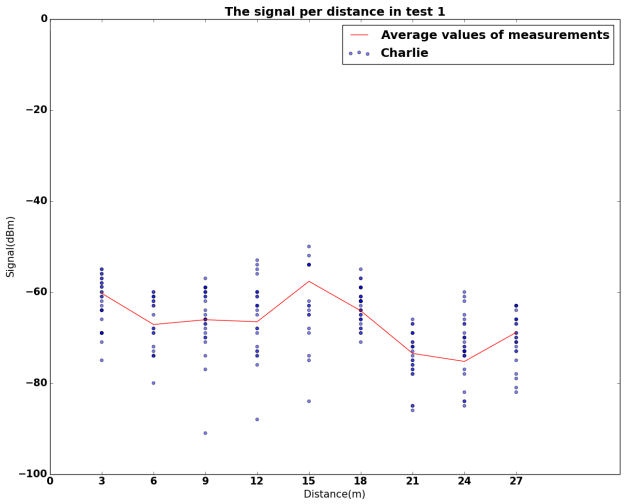


Figure A.9: The determined coordinates by the Fitter algorithm after Test 4

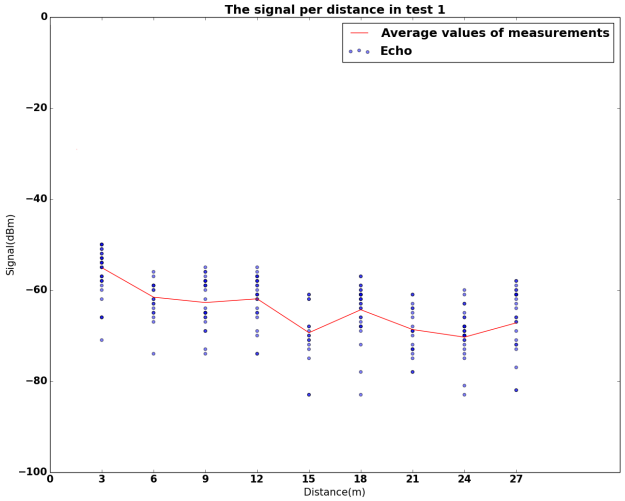
A.5.2 Experiment 2

A.5.2.1 Measurements of the signal strength versus distance-Test 1

Compared to Test 2 and Test 3, Test 1 has fewer measurements performed and it has detections only up to the 27th meter step. The plot of the measurement for this test can be visualized in Figure A.10.



(a) Measurements of Test 1 taken by drone Charlie



(b) Measurements of Test 1 taken by drone Echo

Figure A.10: Measurements of Test 1 from experiment 2

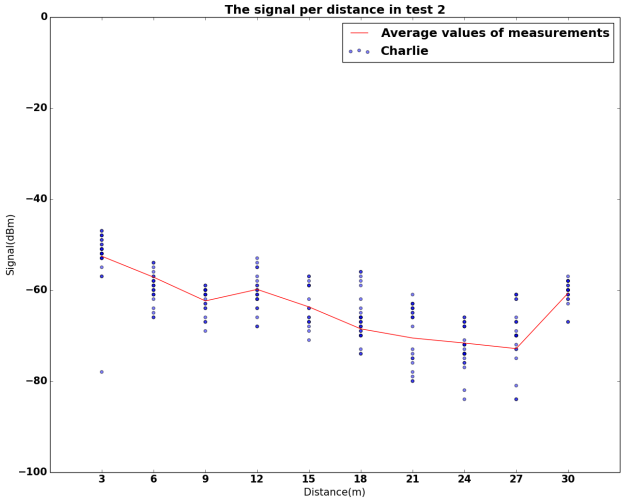
The plots of the measurements of both dronesA.10a and A.10b contain not

only the measurements, but also the average of these measurements at every 3 meter step. It can be seen that the greater the distance is from the drones, the lower the signal strength is.

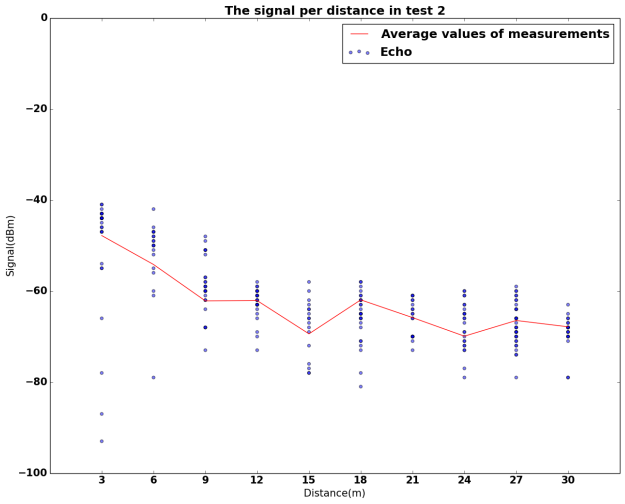
Due to the fact that the drones are placed next to each other, we expect that the signal strength to be detected in the same way, which can be actually seen. However, for the $15th$ meter step the situation is different, as Charlie detects a higher signal than Echo. A reason for this situation may be the shape of the hallway, which at the $15th$ meter step has the entrance of the open-space area, where the desks are located on the left side, while on the right side there are the rooms of other offices and this may attenuate the signal strength.

A.5.2.2 Measurements of the signal strength versus distance-Test 2

By looking at the measurements performed in Test 2 with the results in Figure A.11a and Figure A.11b, it can be seen again that the signal strength decreases the further a cell-phone device is, but this is not the case all the time. If the Charlie and Echo measurements are compared with each other, the shape of their measurements resembles, except for the 3 meter distance. In addition to this, the average value of the measurements decreases up to the 9 meters distance and then it starts increasing in both cases at the 12 meter distance. Then, on one hand, in case of the Charlie drone the average continues to decrease up to 27 meters and then it increases for the 30 meters. On the other hand, this situation does not actually happen for the Echo drone, where the average signal strength increases at the 18 meter distance, followed by another increase but less intense at 27 meters. This seems surprising as we would expect the power of the signal to decrease continuously. One possible reason for this may be the shape of the hallway, the walls, and the metal drawers which may attenuate or may increase the signal strength at different locations.



(a) Measurements of Test 2 taken by drone Charlie

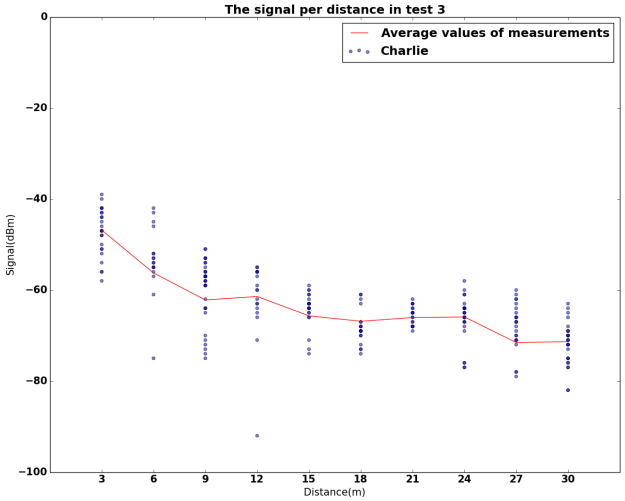


(b) Measurements of Test 1 taken by drone Echo

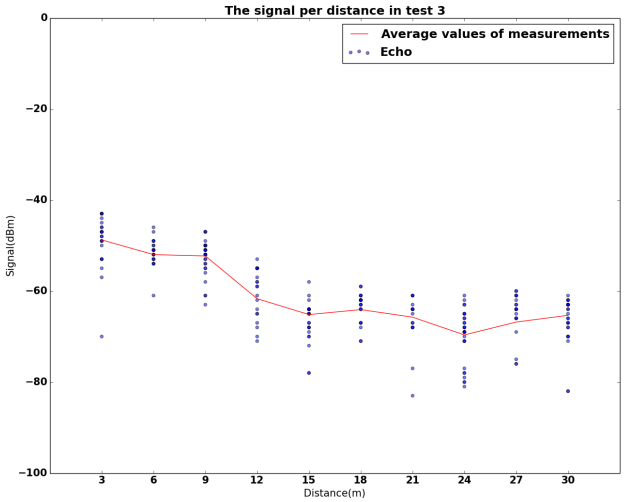
Figure A.11: Measurements of Test 2 from experiment 2

A.5.2.3 Measurements of the signal strength versus distance-Test 3

The measurements of test 3 can be visualized in Figure A.12a and Figure A.12b. Compared to the other experiments, these measurements seem to have in most cases a smoother decrease even though this is not always the case. It can be observed that Charlie has again spikes of the average signal strength at 12, 24, and 30 meters distance, while Echo has an average signal spikes at 9, 18, 27, and 30. The reason why these spikes occur may be due to the shape of the hallway.



(a) Measurements of Test 3 taken by drone Charlie



(b) Measurements of Test 3 taken by drone Echo

Figure A.12: Measurements of Test 3 from experiment 2

A.5.2.4 Chi-square minimization of the average signal strength test 1

As it can be seen in Tables 4.5 and 4.6 in the case of Test 1, the *curve_fit* function found in the *scimpy.optimize* Python package did not manage to find the optimal parameters and instead it returned the initial guesses for several trials. Due to this situation, we decided to try the average value of the parameter estimates obtained by applying the same function for Test 2 and Test 3. Thus,

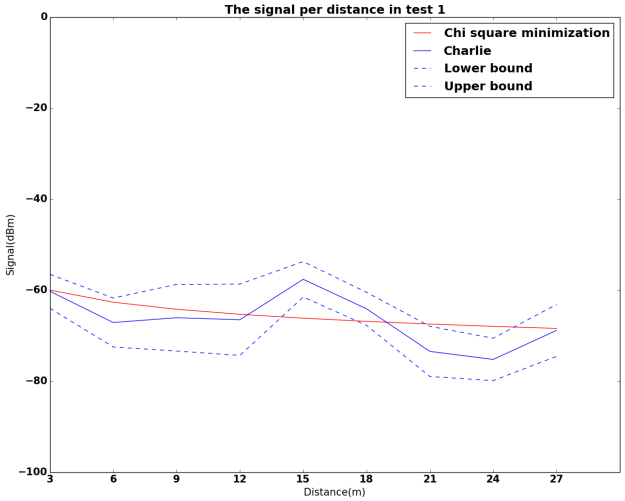
in Figure A.13a and Figure A.13b, the calculated function is plotted, which has the following form:

$$f(distance) = -44.9 - 0.78 \times 20 \times \log_{10} distance$$

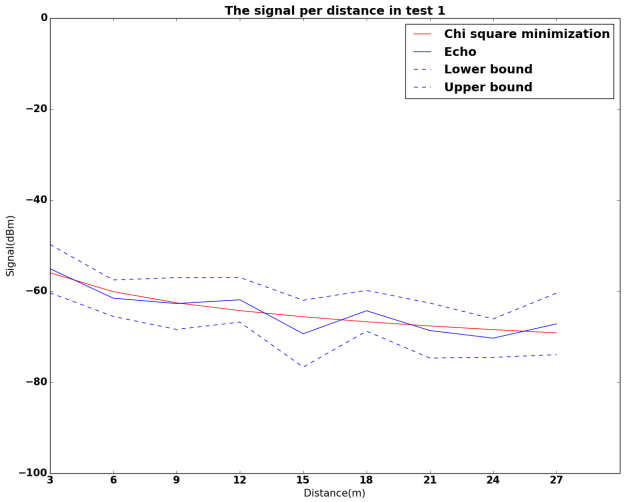
versus the average signal strength obtained from the taken measurements. As it can be seen in Figure A.13a, the fit seems pretty good up to the 12 meter distance. However, in the interval between 12 up to 18 meters, the calculated function does not entirely fit the data. As mentioned before, it may be the situation that the shape of the hallway must have influenced the signal strength. However, this is not the same situation for the Echo drone. According to Figure A.13b, the calculated function, which has the following form :

$$f(distance) = -40.5 - 0.9 \times 20 \times \log_{10} distance$$

seems to fit pretty well the average values of the signal strength. Given the fact that the $curve\_fit$ function did not return any optimal parameters for this data either, because the function may not have converged, these average values of the optimal parameters seem to be quite good, but they may not be the ones where the errors are minimized.



(a) Measurements of Test 1 taken by drone Charlie



(b) Measurements of Test 1 taken by drone Echo

Figure A.13: Measurements of Test 1 from experiment 2

A.5.2.5 Chi-square minimization of the average signal strength test 3

The Figure A.14a and Figure A.14b describe the behavior of the drones Charlie and Echo, respectively in Test 3. The functions are the following:

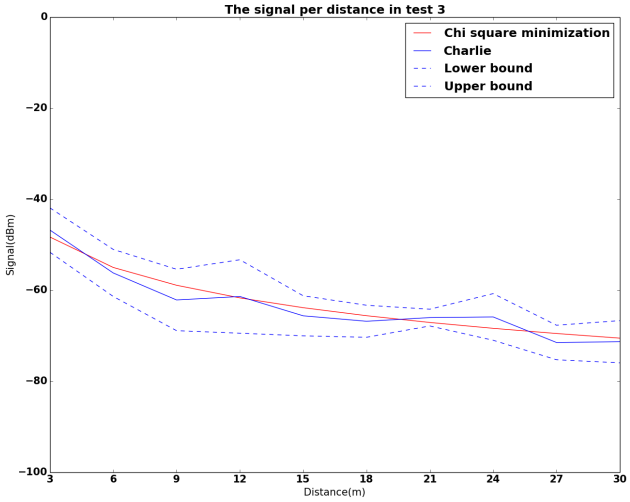
$$f(distance) = -37.7 - 1.1 \times 20 \times \log_{10} distance$$

for the Charlie drone and

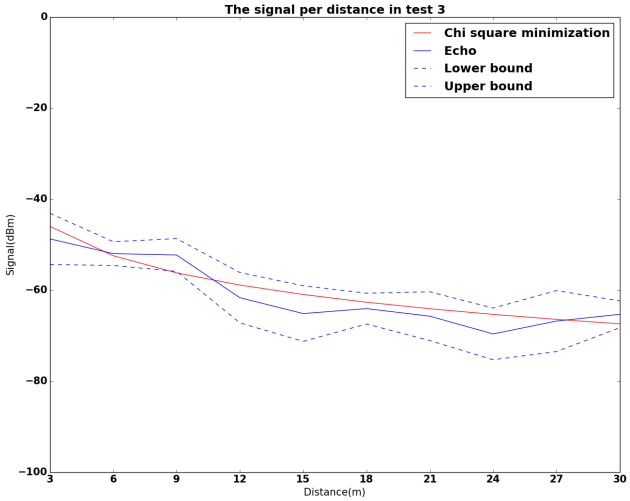
$$f(distance) = -35.8 - 1.1 \times 20 \times \log_{10} distance$$

for the Echo drone.

By looking at the first drone, it can be observed that the determined function with the optimal parameters seems to fit the data better than in the previous tests. The same thing seems to be for the Echo drone, even though it can be seen that the function fits instead the lower bound of the average signal strength at the 9 meter distance.



(a) Measurements of Test 3 taken by drone Charlie



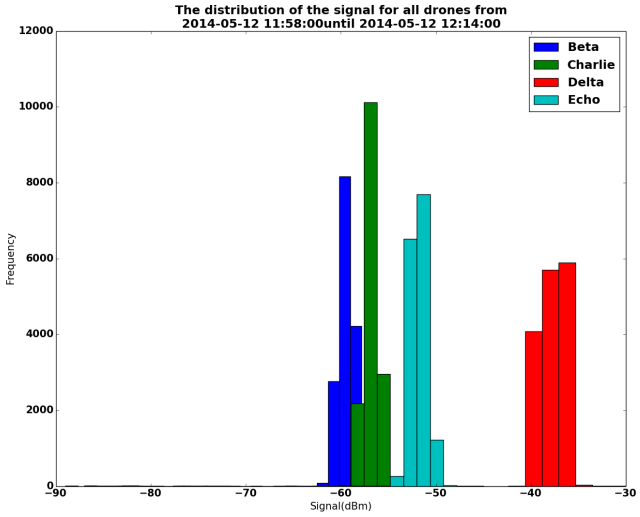
(b) Measurements of Test 3 taken by drone Echo

Figure A.14: Measurements of Test 3 from experiment 2

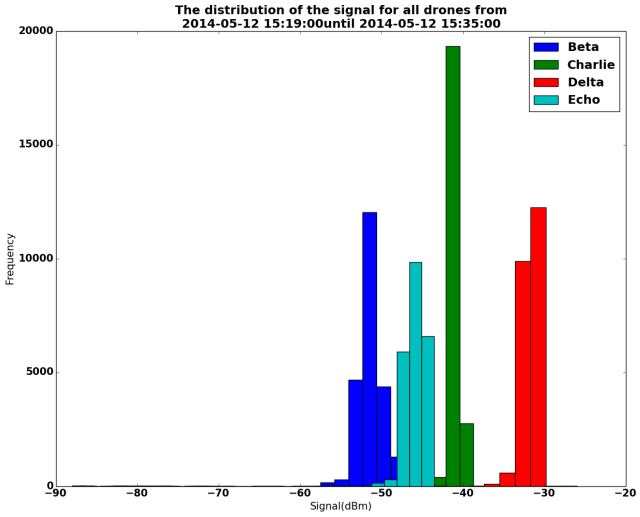
A.5.3 Experiment 3

A.5.3.1 Test 2

The distributions of the signal strengths can be visualized in Figure A.15a and A.15b. On one hand, as seen in Test 1, the highest average signal strength can be also seen for Delta for both devices. Apparently, this time the iPhone seems to detect an higher average than the HTC for Delta drone. In addition to this, the iPhone seems to communicate more with the drones than the HTC when streaming data. On the other hand, the Beta drone has the lowest average signal strength in both cases with -59.9 ± 0.9dBm for the HTC versus -51.4 ± 1.7dBm, respectively for the iPhone. The main difference between the two devices is that the HTC is recorded on average better by Echo than Charlie in contrast to the iPhone, which is detected better by Charlie and less by Echo.



(a) HTC mobile device

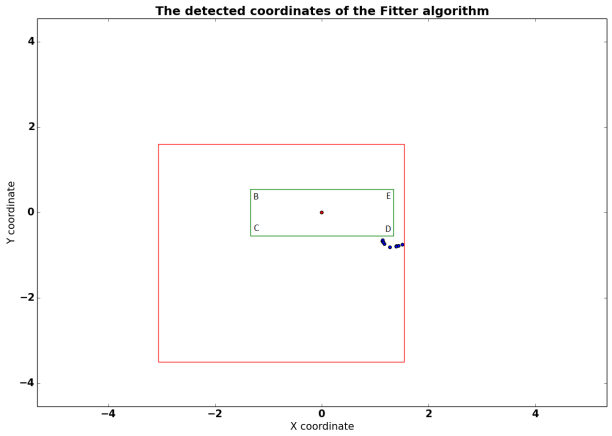


(b) iPhone mobile device

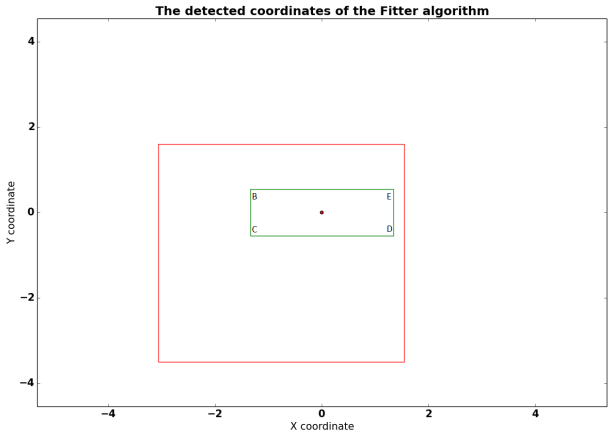
Figure A.15: The distributions of the signal strength of Test 3 from experiment 2

The results of test 2 are not as encouraging as the ones obtained in test 1, because the number of calculated coordinates is small(10 for the HTC and 0 for the iPhone) given the high number of sniffed packets. However, a positive aspect may be that in the HTC case, these points are clustered and the mobile device is still seen in the area of the room where the experiment was performed. The values of the average x-coordinate are still pretty small 1.3 ± -0.7 and -0.7 ± 0.1 for the y coordinate, respectively. These values are higher compared

to the ones of the previous experiment.



(a) HTC mobile device

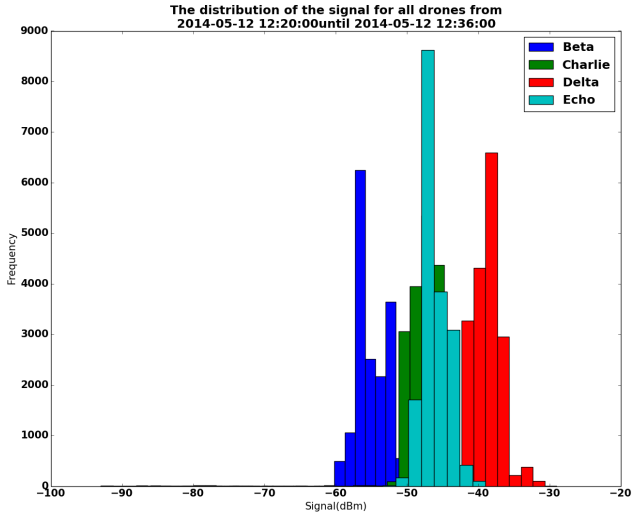


(b) iPhone mobile device

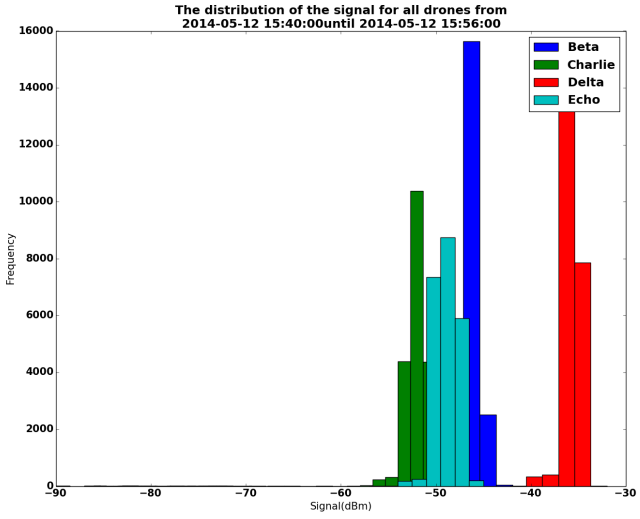
Figure A.16: The coordinates detected by the Fitter algorithm in case of Test 2 from experiment 3

A.5.3.2 Test 3

According to Figures A.15b and A.17b, the Delta drone has again the highest values of the average signal strength in the case of both mobile devices. Moreover, in the case of the iPhone mobile device, the second highest average signal strength is followed by the Beta drone, then by the Echo drone, and finally by Charlie. This is not the case with the HTC, which is followed by Echo, Charlie, and Beta lastly. Besides this, the distributions of Beta, Charlie, and Echo overlap.



(a) HTC mobile device



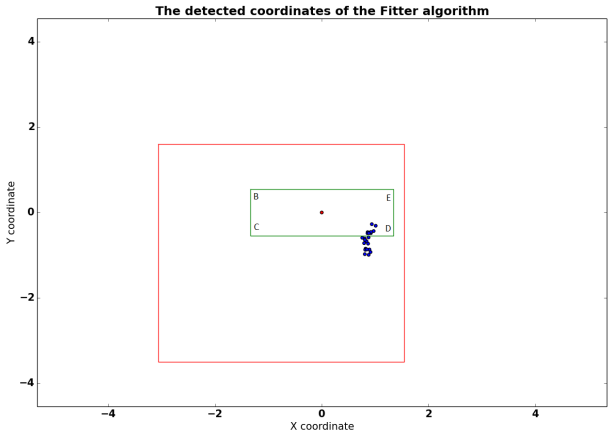
(b) iPhone mobile device

Figure A.17: The distributions of the signal strength of Test 3 from experiment 2

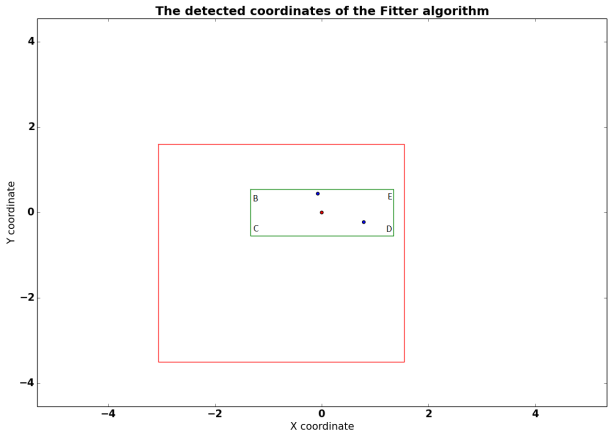
According to table4.9, the number of calculated points by the Fitter algorithm is 25 for the HTC and 2 for the iPhone. This result is comparative to the previous test, as the algorithm could not compute the position of the iPhone device more than twice, but these points were still situated on the table. However, an interesting outcome was again that the determined points were located close to the Delta drone for the HTC case.

The average coordinates obtained by the algorithm in the HTC case were

0.9 ± 0.1 for the x-coordinate and -0.7 ± 0.2 for the y-coordinate, while, in the iPhone's case, they were 0.4 ± 0.4 for the x-coordinate 0.1 ± 0.3 for the y coordinate, respectively.



(a) HTC mobile device

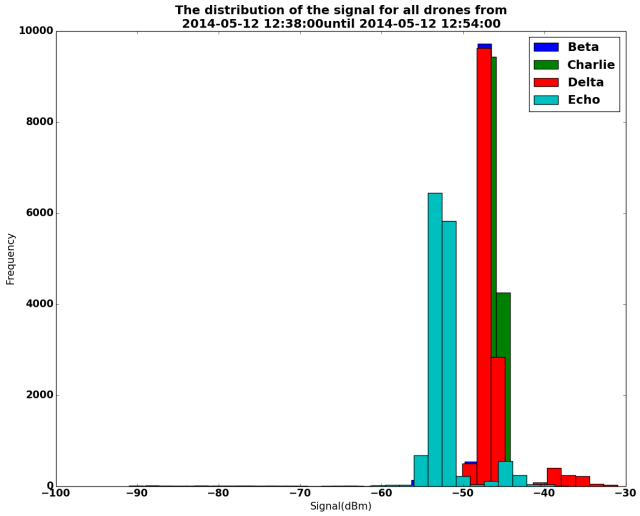


(b) iPhone mobile device

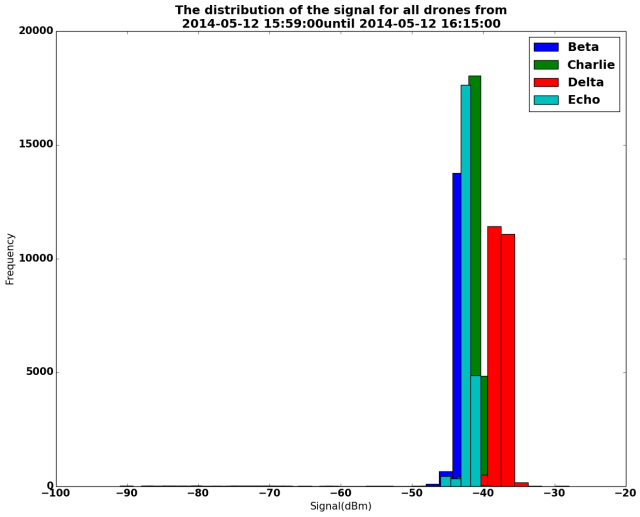
Figure A.18: The coordinates detected by the Fitter algorithm in case of Test 3 from experiment 3

A.5.3.3 Test 4

In the last performed test, it can be seen in Figure A.19a that the distributions of Beta, Charlie, and Delta overlap. However, Echo seems to be shifted to the left side and has the smallest average detected signal strength compared to the rest. The same situation happens for the iPhone where the distributions of Beta, Charlie, and Echo overlap, which is in contrast to the HTC case, where the distribution of Delta is shifted to the right side and has a higher signal strength compared to the other ones.



(a) HTC mobile device

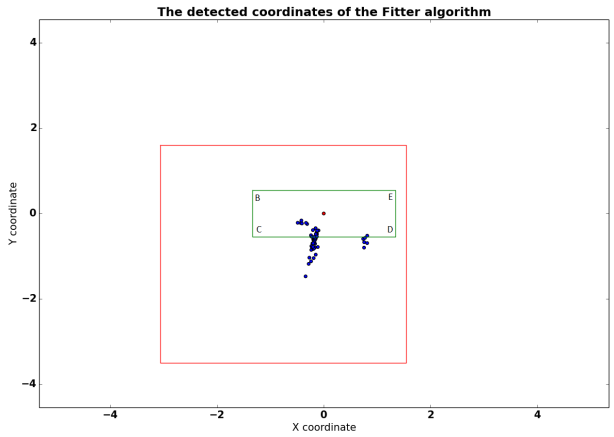


(b) iPhone mobile device

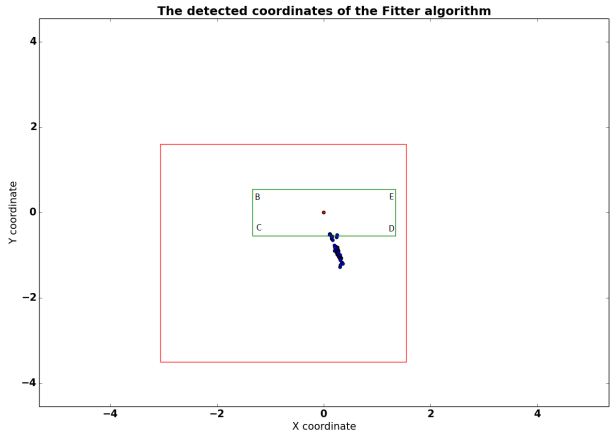
Figure A.19: The distributions of the signal strength of Test 4 from experiment 3

According to Figures A.20a and A.20b, it can be clearly seen that test 4 represents by far one of the best results obtained from this experiment as the number of detected points is 63(HTC) and 63(iPhone). Each test was performed for 16 minutes. Normally, at every 10 seconds, the Fitter algorithm should provide at least one point. This would mean that for these 16 minute experiments around $\frac{(16\times 6)}{10} = 96$ points should be determined. This means that the algorithm performed unexpectedly well in the iPhone case.

On the other hand, the average coordinates in the HTC case are -0.1 ± 0.3 and -0.6 ± 0.2 versus the ones calculated for the iPhone with an average for x equal to 0.3 ± 0.0 and -0.9 ± 0.1 for y, respectively.



(a) HTC mobile device



(b) iPhone mobile device

Figure A.20: The coordinates detected by the Fitter algorithm in case of Test 4 from experiment 3

A.6 The performance of the Fitter algorithm - Experiment 1

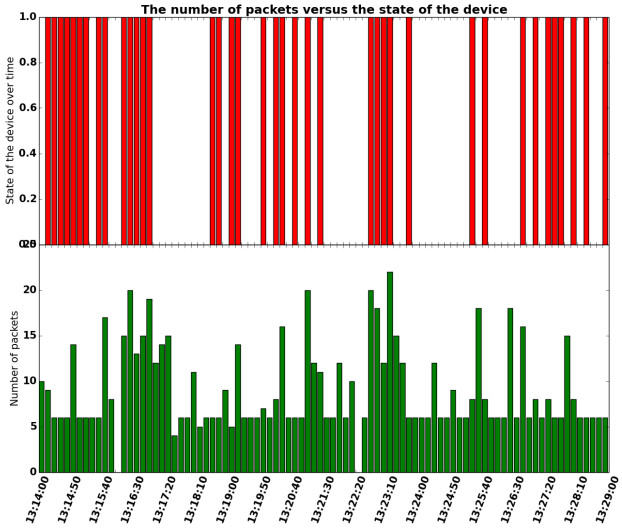
A.6.1 Test 2

In Test 2, we remind the reader that the position of the iPhone mobile device was located with the home button towards the Charlie and Foxtrot routers. According to Table 5.1, the number of calculated points by the algorithm is

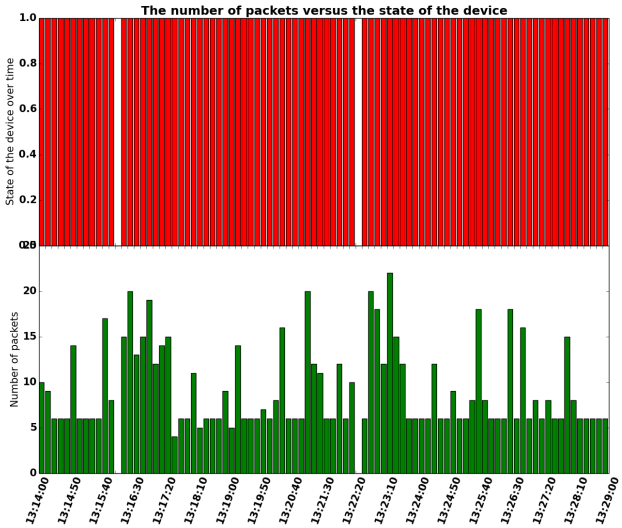
39, which is much less compared to the previous test. Thus, there were 49 time intervals of 10 without no detections and no pattern drawn from their non-detections. However, after the bug fix, this issue was remedied and all the time intervals with packets had points computed. For this experiment, 2 time intervals seem to not have any calculated points for the entire experiment. Therefore, the probability of calculating points by the algorithm is equal to 1, as in all the cases where packets were available it computed them.

Figure A.21 shows a nice shape in the number of packets over time. The majority seem to be more than 5 packets per time interval of 10 seconds. This is reassuring, because this experiment was performed mainly by pressing the home button once a minute. This may indicate that even though the transition between idle to active state is performed, the mobile devices still send sufficient packets which can be used for applying the trilateration technique.

When looking at the plots of the coordinates Figure A.22, the calculated points by the algorithms can be visualized before the bug fix and after. As seen in the previous test, the number of points has definitely increased, but the performance of the resolution did not. In Figure A.22b, the points are scattered in all directions, even though most of them are located on the right side of the reference point. One of the possible explanations for this result may be that the routers that seem to record a higher signal strength compared to the others are Foxtrot and Delta. Compared to Figure A.22a, Figure A.22b contains much more outliers, which are especially located in outside the area of the table and some of them even outside the area of the room. This appearance of new points affected as mentioned before the average values of x and y. For example, the average value of the x coordinate was 0.4 ± 1 while after the bug fix it became -1.3 ± 2 which represents a significant difference. In addition to this, the average y coordinate was initially -0.1 ± 0.7 and later was -1.2 ± 3.7. Thus, not only the average values of both coordinates are further away from 0, but their uncertainties have increased significantly.

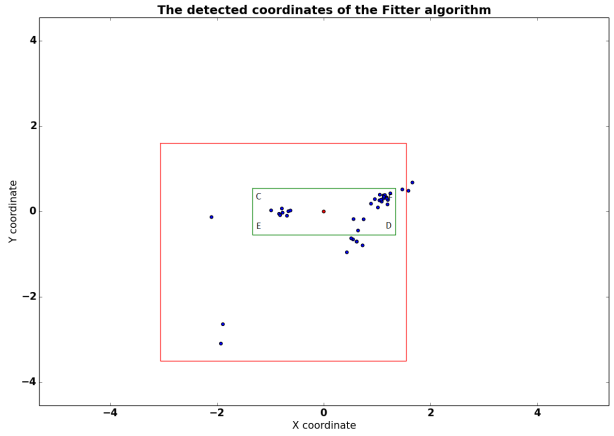


(a) The comparison between the states and data packets experiment 1 Test 2(before)

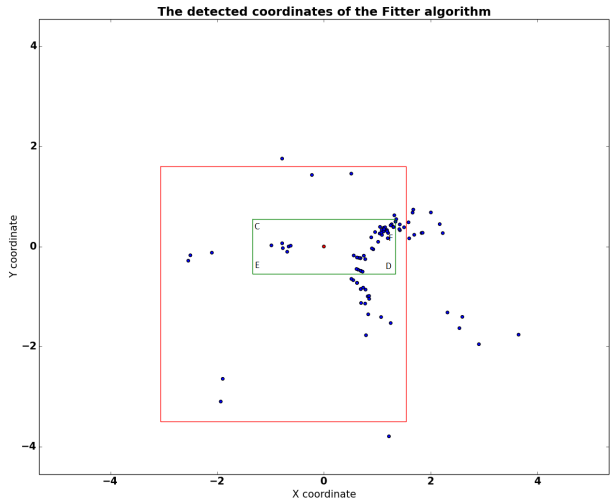


(b) The comparison between data records and packets experiment 1 Test 2(after)

Figure A.21: The comparison between data records and packets experiment 1 Test 2(before and after)



(a) The calculated coordinates by the Fitter algorithm experiment 1 Test 2(before)



(b) The calculated coordinates by the Fitter algorithm experiment 1 Test 2(after)

Figure A.22: The calculated coordinates by the Fitter algorithm experiment 1 Test 2(before and after)

A.6.2 Test 3

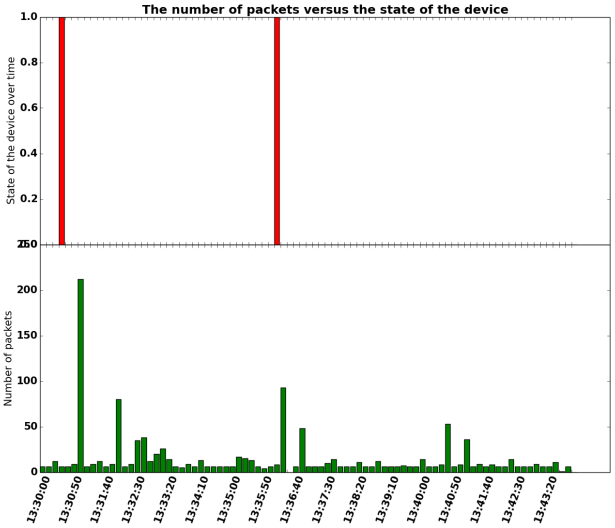
In Test 3, the position of the mobile device with the home button towards the Foxtrot and Delta routers. One would expect that the routers that would detect the highest signals strengths would be Charlie and Echo. This is not entirely true as the Foxtrot and Delta seem to again dominate in having the largest recorded signal strengths. Another particularity for this test was that the algorithm initially computed only 2 points, which did not make any sense as there were

sufficient number of packets for every 10 second time intervals. Thus, according to Table 5.1, there were 81 time intervals of 10 seconds with no detections. In addition to this, the percentage of time intervals that had detections is $2.4\% \pm 5.8\%$, which is low given the $97.6\% \pm 5.8\%$ efficiency of the WiFi tracking system, as there were 2 time intervals with no received packets. However, after the bug fix the algorithm computed points for all the time intervals and the total efficiency increased significantly from $2.4\% \pm 5.8\%$ to $97.6\% \pm 5.8\%$.

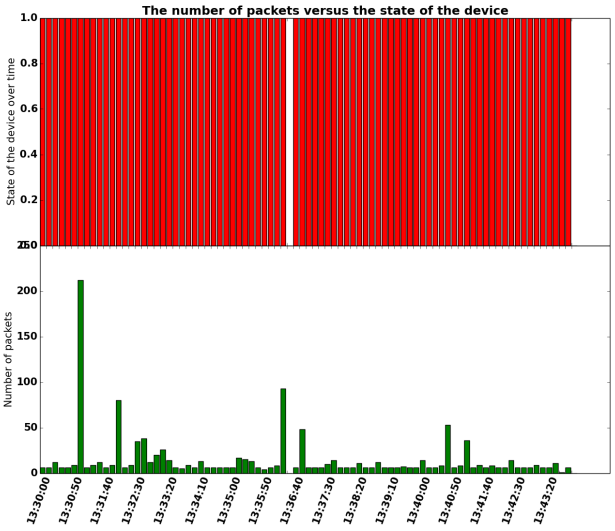
According to the plots of the coordinates in Figures A.24a and A.24b[1] , it can be seen a significant difference between the two of them, as an increased number of points are plotted after the correction of the algorithm. All these points seem to be located as in Test 2 on the right side of the reference point. The majority of points are close to the Foxtrot and Delta routers, which had the highest signal strengths. This may reveal again that the orientation of the device has an impact on the signal strength which significantly influences the points calculated by the algorithm. Thus, by correcting the signal strength the performance of the points may be improved. This represents a reassuring result as correcting the signal strength can be controlled, but the position of the mobile device cannot.

As mentioned in other tests as well, even though the number of points increased significantly, the performance has decreased. For example after the bug fix, the average x coordinate was 71.6 ± 50.4, while the average y coordinate was -5.1 ± 41.1. Thus, a search into the dataset of the coordinates was performed which revealed the existance of 3 computed points with very high x and y coordinates, which have a small probability of appearance. However, this situation may provide evidence that further corrections should be made to the algorithm such that these high values are avoided from analyses.

[1]It may be the case that the plot of the states and packets after the bug fix(Figure A.23b) does not reveal the second time gap. This may happen, because the matplotlib package for python sometimes does not show the gaps due to the multitude of 10 second time intervals. Thus, it shrinks the image and some time intervals overlap with each other.

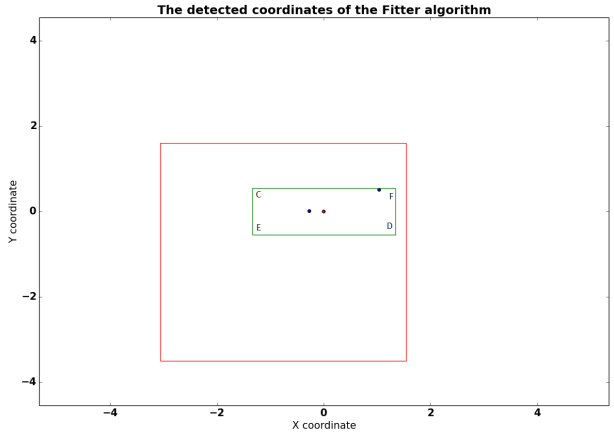


(a) The comparison between the states and data packets experiment 1 Test 3(before)

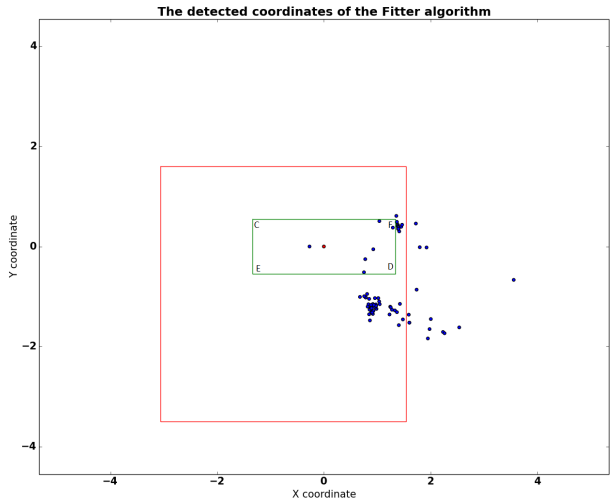


(b) The comparison between the states and data packets experiment 1 Test 3(after)

Figure A.23: The comparison between the states and data packets experiment 1 Test 3(before and after)



(a) The calculated coordinates by the Fitter algorithm experiment 1 Test 3(before)



(b) The calculated coordinates by the Fitter algorithm experiment 1 Test 3(after)

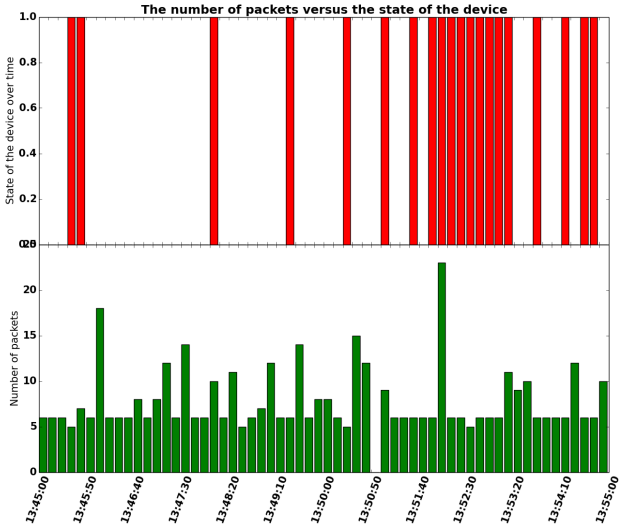
Figure A.24: The calculated coordinates by the Fitter algorithm experiment 1 Test 3(before and after)

A.6.3 Test 4

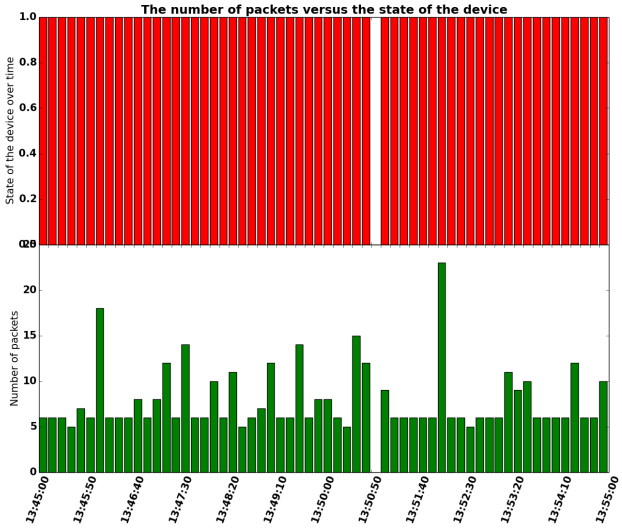
The fourth test of experiment 1 was with the home button positioned towards the Echo and Delta routers. When looking at the Figures A.25a and A.25b, it can be seen that on average there are 5 packets per 10 second time intervals, except for one time interval which does not contain any data packets. Thus, the efficiency of the WiFi tracking system is $98.3\% \pm 5.8\%$. This is a significant result as this experiment tested the communication with the probe packets.

Initially, for this test only 20 points were detected, but afterwards the algorithm computed 39 more points which represented all the number of 10 second time intervals with packets, but no calculated points.

When looking at the Figure A.26, it can be seen again that the newly computed points do not improve the accuracy of the detections, as most of them are outliers are located either in the area of the room or outside the room. Because the Foxtrot had on average the highest detected signal strength, the majority of the points are located close to this router. In addition to this, there are also points scattered on the table, which seem to have the x coordinate close to 0. However, on an aggregated level, the average x coordinate is 0.5 ± 0.8, while the y coordinate is 0.7 ± 1.5. Thus, as expected, the resolution of the average coordinate did not improve even though there were points computed.

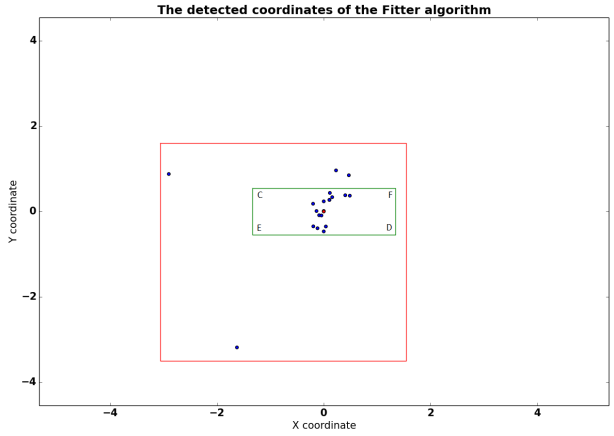


(a) The comparison between the states and data packets experiment 1 Test 4(before)

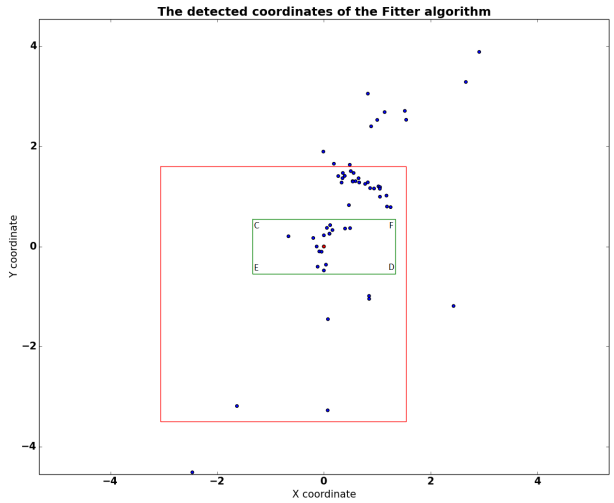


(b) The comparison between the states and data packets experiment 1 Test 4(after)

Figure A.25: The comparison between the states and data packets experiment 1 Test 4(before and after)



(a) The calculated coordinates by the Fitter algorithm experiment 1 Test 4(before)



(b) The calculated coordinates by the Fitter algorithm experiment 1 Test 4(after)

Figure A.26: The calculated coordinates by the Fitter algorithm experiment 1 Test 4(before and after)

A.7 The performance of the Fitter algorithm - Experiment 3

A.7.1 Test 2

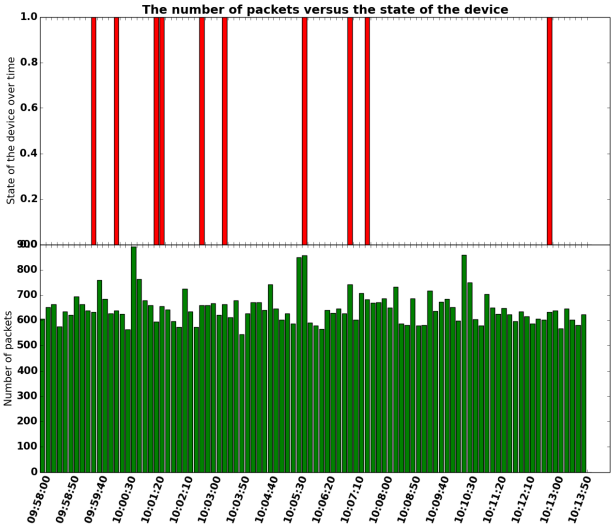
Test 2 was performed by having the mobile devices with their home buttons oriented towards routers Beta and Echo. According the results of test 2(Subsubsection

A.5.3.1), the average signal strengths could be found for both devices in the case of the Delta drone, followed by Echo and Charlie in the HTC case and the reverse order the iPhone case. In addition to this, this test together with test 3 provided initially poor results with respect to the number of points being calculated. However, this result changed after the fix and the number of compute points increased from 10(HTC) and 0(iPhone) to 95 points(Figure A.27 and Figure A.29). In addition to this, the algorithm could not only compute the coordinates for one time interval for both mobile devices.

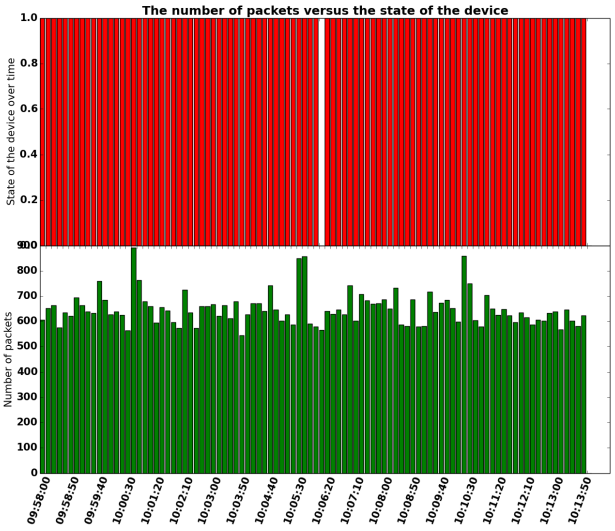
A.7.1.1 HTC

An interesting result in the HTC case was that the average x and y coordinates did not change even though the number of computed points was significantly higher after the bug fix. This was a positive result compared to the one obtained in experiment 1 where the average x and y changed considerably. For example, according to Table 5.2, the average previous x coordinate was 1.3 ± 0.1, while the current one is 1.3 ± 0.1, while for the y coordinate the average was -0.7 ± 0.1 compared to -0.8 ± 0.0.

Figures A.28a and A.28b show the points computed by the algorithm. As it can be seen in the first figure, there is a small cluster of points located in the vicinity of the Delta router. As mentioned before, the Delta router recorded the highest signal strength out of the four routers and this may represent one of the reasons why the cluster of points is located in that area. A positive aspect is that this cluster of points is still maintained after the bug fix(Figure A.28b), which seem to be concentrated into the same space with a similar shape. Besides this, it can be observed that the points are still in the area of the room and close to the area of the table. This may indicate that if the routers are properly calibrated the results may improve.

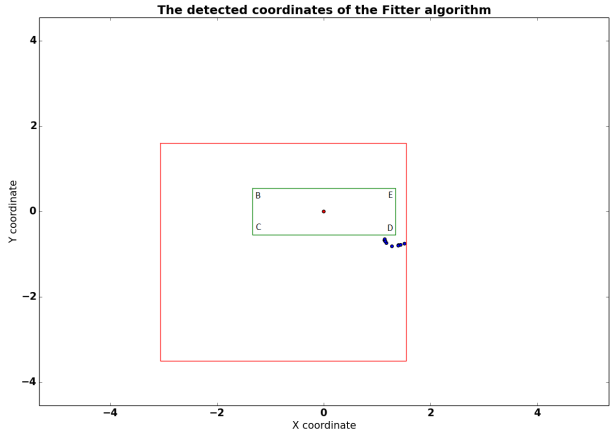


(a) The states and coordinates detected by the Fitter algorithm for the HTC mobile device experiment 3 Test 2(before)

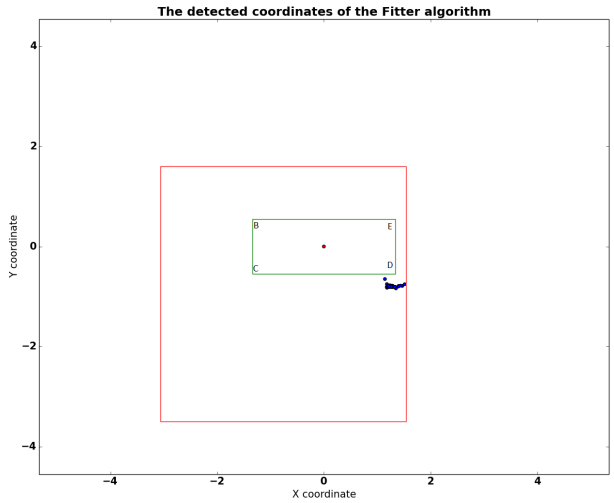


(b) The states and coordinates detected by the Fitter algorithm for the HTC mobile device experiment 3 Test 2(after)

Figure A.27: The states and coordinates detected by the Fitter algorithm for the HTC mobile device in experiment 3 Test 2(before and after)



(a) The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 2(before)



(b) The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 2(after)

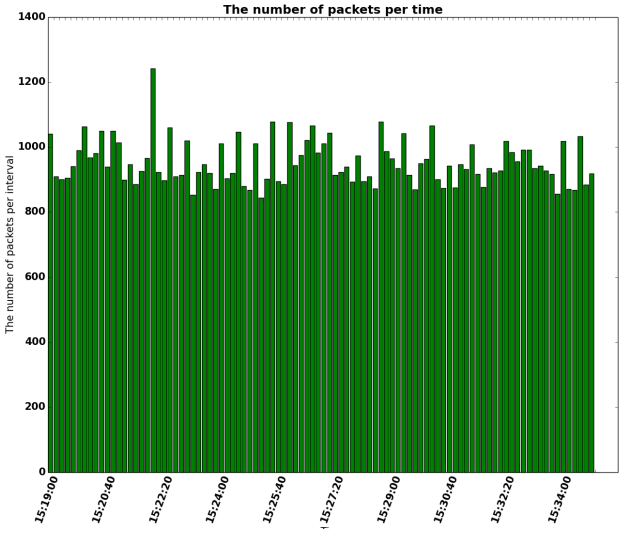
Figure A.28: The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 2(before and after)

A.7.1.2 iPhone

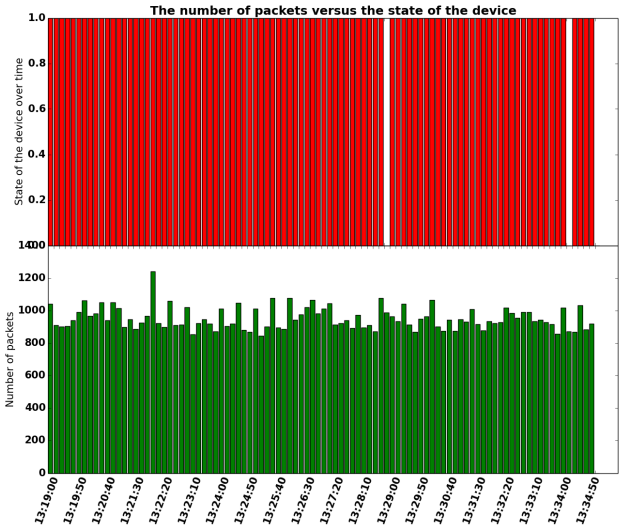
Test 2 for the iPhone was initially the one which performed the worst. This was unexpected as there seemed to be a lot of data packets for each 10 second time interval. This test was similar to other tests from a number of data packets perspective, but later contained data points compared to test 2 which did not compute any data point. However, after the bug fix, for this test 95 data points were computed. Due to the absence of data points, there cannot be

made any comparison before the initial situation and the current situation from a resolution of coordinates point of view. It can be affirmed that the current average x coordinate is 1.2 ± 2.2 and -1.2 ± 1.0 for the y coordinate, respectively.

In Figure A.30b, a cluster of points can be observed close to the Delta drone and several scattered points are located between the Charlie and the Delta drones. These points are not on the table, but inside the room. This may indicate that the signal strength plays a key role in the way the coordinates are reconstructed(Subsubsection A.5.3.1).

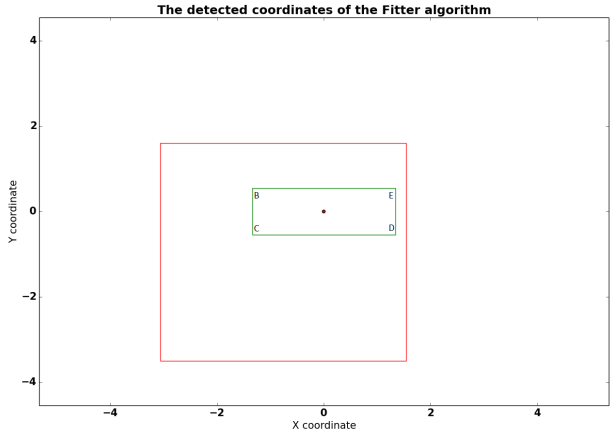


(a) The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 2(before)

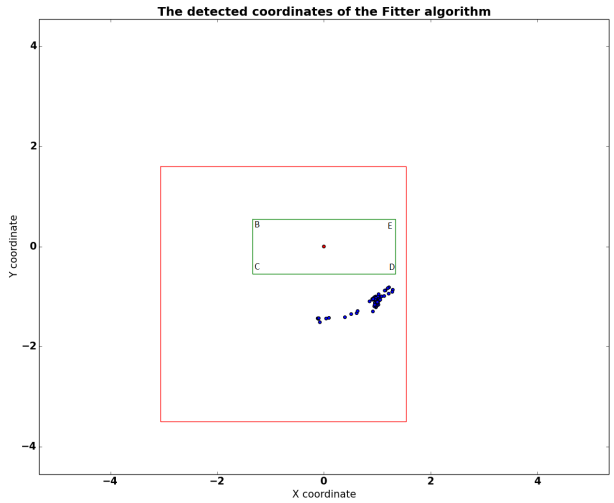


(b) The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 2(after)

Figure A.29: The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 2(before and after)



(a) The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 2(before)



(b) The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 2(after)

Figure A.30: The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 2(before and after)

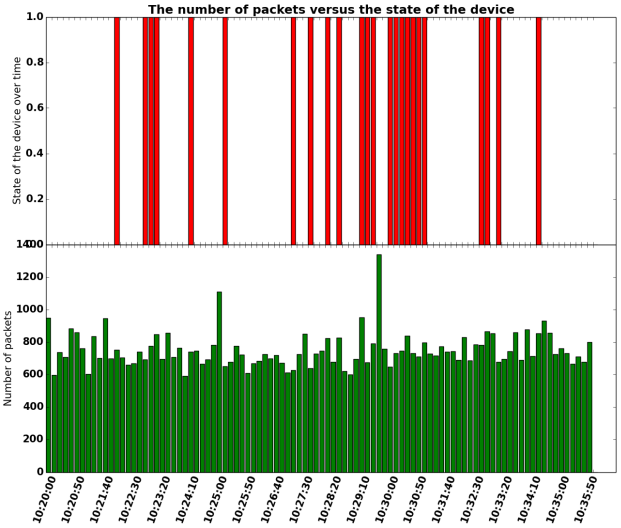
A.7.2 Test 3

Besides Test 2, Test 3 also seemed to have a poor performance initially for both mobile devices, actually the iPhone performed worse from a number detected data points) However, this situation changed when the algorithm was fixed and the number of points computed for the iPhone was 95 compared to 94 in the HTC case which had one time interval for which the algorithm could not provide a solution even though the number of packets was different than 0. In Tables 5.2

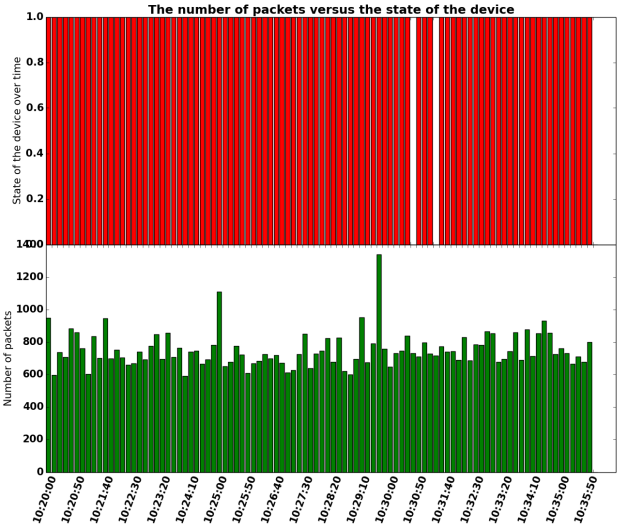
and 5.3, the average x and y coordinates can be visualized. As it can be seen, the average coordinates in the HTC case did not seem to increase significantly, compared to the iPhone case where the x coordinate increased slightly from 0.4 ± 0.4 to 1.0 ± 0.2 and 0.1 ± 0.3 to -0.9 ± 0.3 for the y coordinate, respectively.

A.7.2.1 HTC

The plots of the coordinates can be observed in Figure A.32 where there is a formed cluster of points which is situated close to the Delta router. This cluster is split between the area of the table and the area of the room. Between the Figure A.32a and Figure A.32b, there seems to be a difference only from a number of computed points perspective. This is also supported by the fact that the average x and y coordinates changed slightly from the previous dataset to the new obtained one after the algorithm was fixed. As mentioned before, the position of the cluster may be influenced by the signal strengths of the Delta and Echo routers.

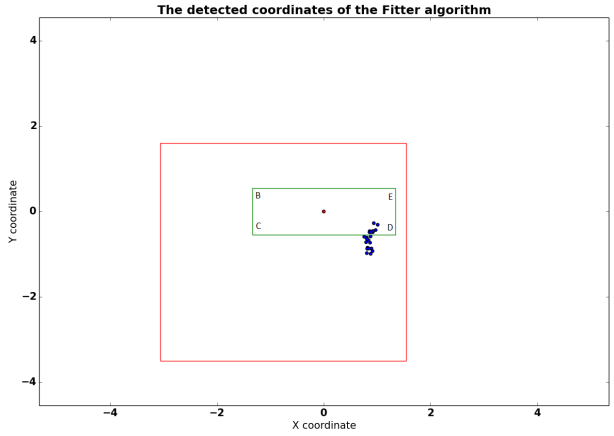


(a) The states and coordinates detected by the Fitter algorithm for the HTC mobile device experiment 3 Test 3(before)

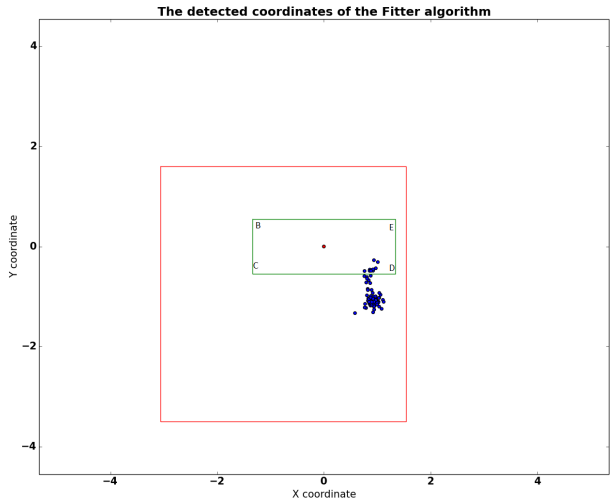


(b) The states and coordinates detected by the Fitter algorithm for the HTC mobile device experiment 3 Test 3(after)

Figure A.31: The states and coordinates detected by the Fitter algorithm for the HTC mobile device experiment 3 Test 3(before and after)



(a) The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 3(before)



(b) The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 3(after)

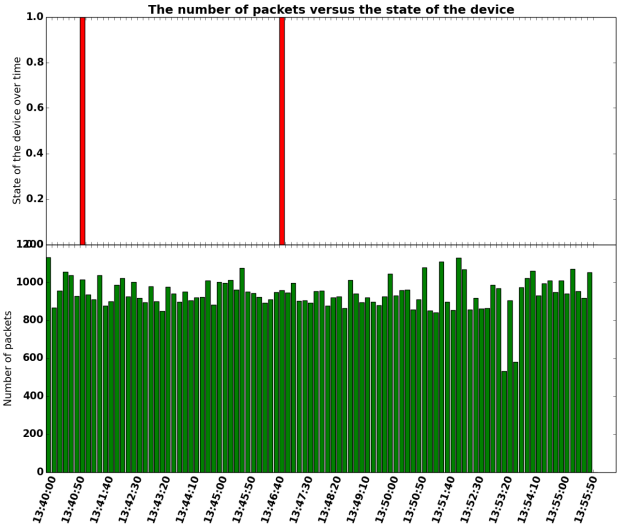
Figure A.32: The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 3(before and after)

A.7.2.2 iPhone

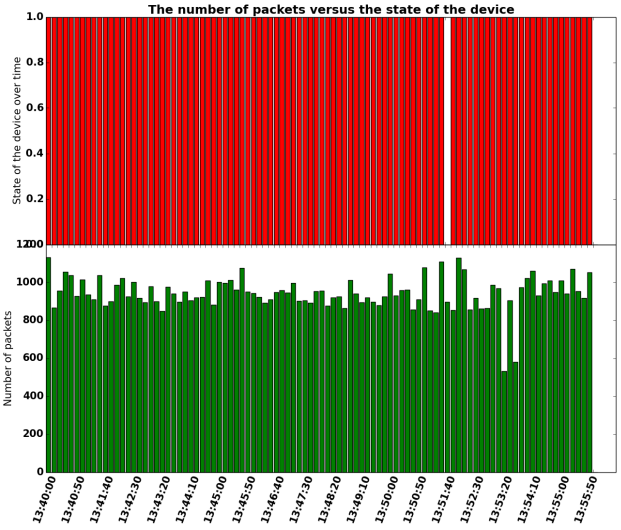
The number of computed points for this test was significantly increased from 2 to 95 which meant that the probability that the algorithm determined points raised from $2.1\% \pm 5.8\%$ to $99.0\% \pm 3.5\%$(Table 5.3).

The plots of the points computed by the algorithm can be visualized in Figure A.34 where two main clusters of points can be seen. On one hand, there is the one which is close to the Delta router and which is located at the area

of the table. These points seem to have smaller x and y coordinates. On the other hand, there is another area with scattered points that are located inside and outside the area of the room. These points are positioned in the middle between the Beta and Echo routers. This result may be explained by the fact that the distributions of the signal strength of the Beta and Echo routers seem to overlap with each other.

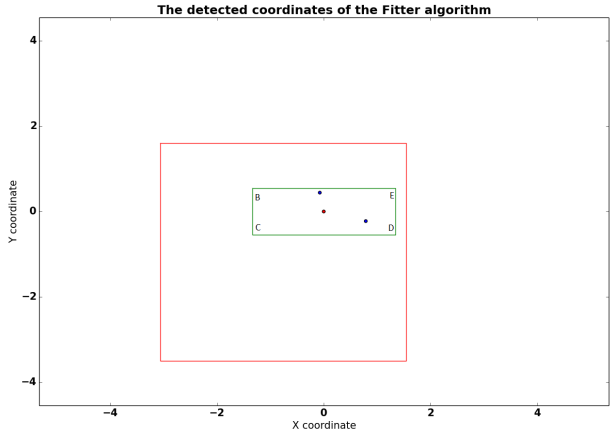


(a) The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 3(before)

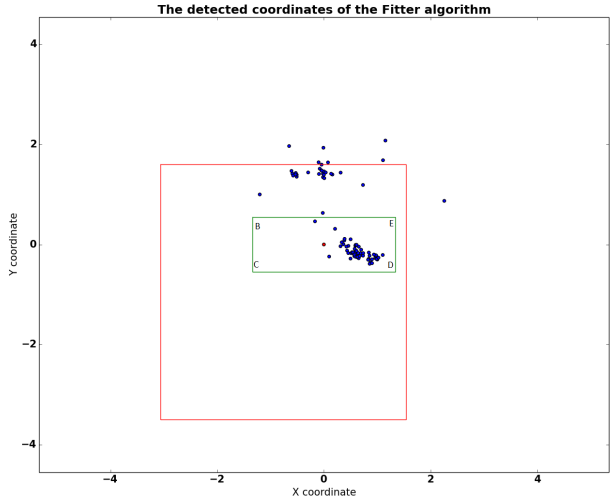


(b) The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 3(after)

Figure A.33: The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 3(before and after)



(a) The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 3(before)



(b) The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 3(after)

Figure A.34: The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 3(before and after)

A.7.3 Test 4

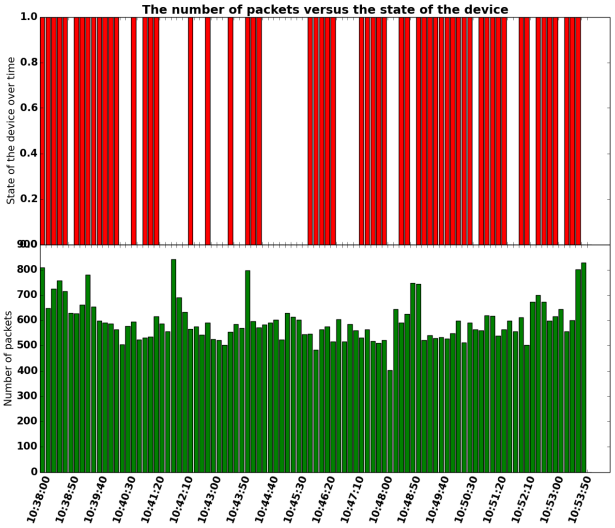
Test 4 was performed with the mobile devices having their home button oriented towards Charlie and Delta routers. From a performance perspective, this test can be compared with Test 1 which also had sufficient initial points computed. Thus, it can be seen in Tables 5.2 for the HTC and 5.3 for the iPhone that the average x and y coordinates did not considerably alter. This seems a positive aspect in the sense that the algorithm is able to be consistent even though some

changes were made to it. From an efficiency point of view, the total efficiency the tracking system increased from $64.9\% \pm 16.8\%$ to $96.9\% \pm 6.1\%$ in case of the HTC, but, in the iPhone case, it only changed slightly from $95.9\% \pm 7.0\%$ to $96.9\% \pm 6.1\%$.

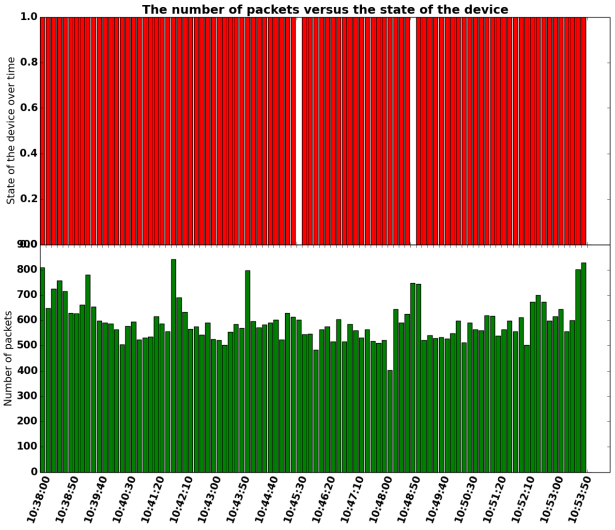
A.7.3.1 HTC

Test 4 of the HTC revealed that only 63 points were computed initially. However, this represented the best result obtained for this mobile device. After the algorithm was modified, the final number of computed points was 94 which was equal to the one obtained for the iPhone. From a coordinates' perspective, there was no significant modification compared to the previous results. For example, the initial average x coordinate was equal to -0.1 ± 0.3 and -0.6 ± 0.2 for y, while the new ones are -0.2 ± 0.3 for x and -0.9 ± 0.5 for y.

According to Figure A.36, one can see two main areas with points. One of them seems to have a vertical straight shape slightly positioned to the left and downward side of the reference point. These points tend to have the x coordinate close to 0. In addition to this, they are split between the area of the table and the area of the room and they are located between Delta and Charlie routers. This may be explained by the fact that the routers with the highest signal strength were Delta, followed by Charlie, Echo, and Beta which had overlapping distributions of the signal strength.

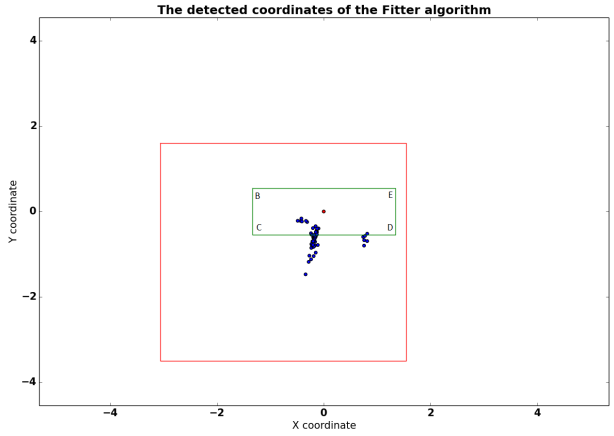


(a) The states and coordinates detected by the Fitter algorithm for the HTC mobile device in experiment 3 Test 4(before)

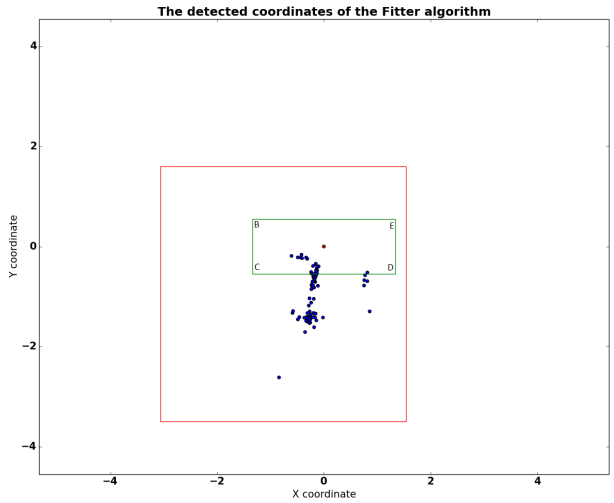


(b) The states and coordinates detected by the Fitter algorithm for the HTC mobile device in experiment 3 Test 4(after)

Figure A.35: The states and coordinates detected by the Fitter algorithm for the HTC mobile device in experiment 3 Test 4(before and after)



(a) The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 4(before)



(b) The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 4(after)

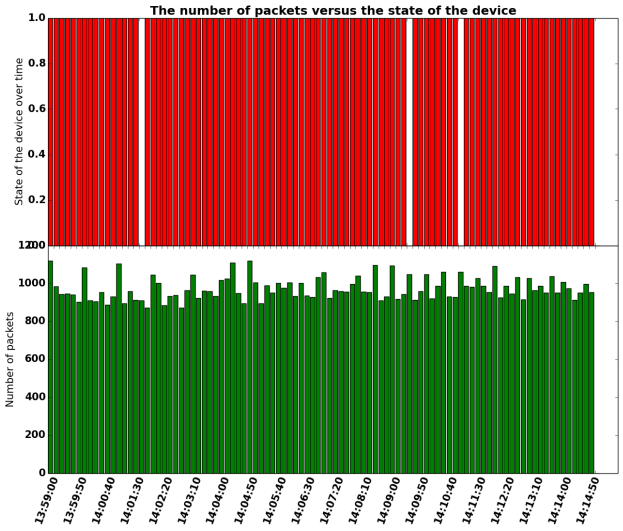
Figure A.36: The reconstructed coordinates by the Fitter algorithm for the HTC mobile device experiment 3 Test 4(before and after)

A.7.3.2 iPhone

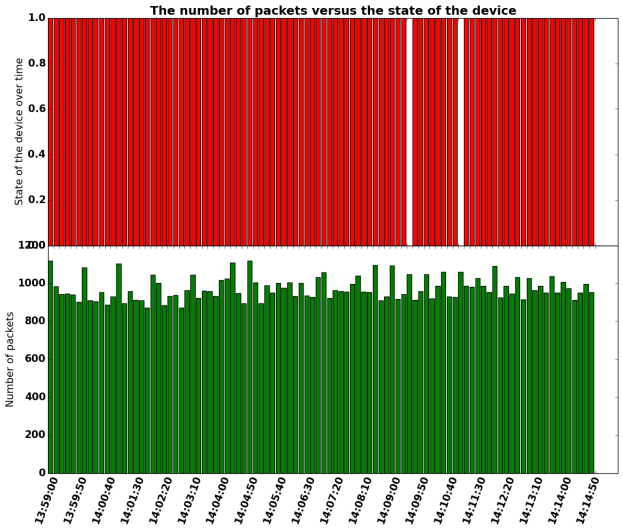
This test contained an unexpectedly large number of computed data points compared to the rest(93 points). Thus, there only 3 time intervals of 10 seconds which did not contain a reconstructed data point by the algorithm. For 2 out of these three 3 time intervals, the algorithm after it was repaired it was able to compute points(Figure A.37). This was an encouraging result

Figure A.38 reveal the plots of the reconstructed points for this mobile de-

vice. As expected, the shape of the plot did not change, as only two additional points where added. This shape seems to have a vertical straight form located between the Charlie and Delta routers and outside the area of the table, which may be explained by the overlapping distributions of the Charlie and Delta routers.

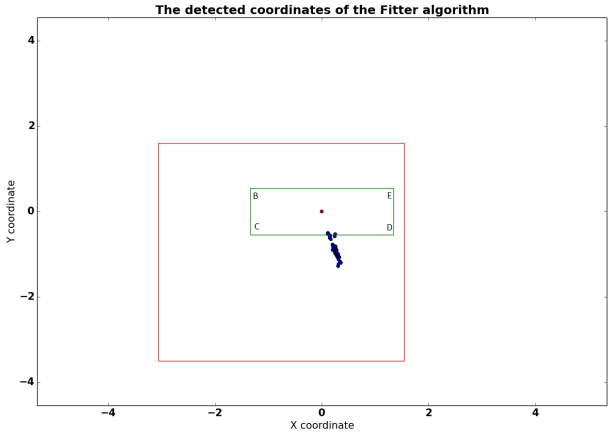


(a) The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 4(before)

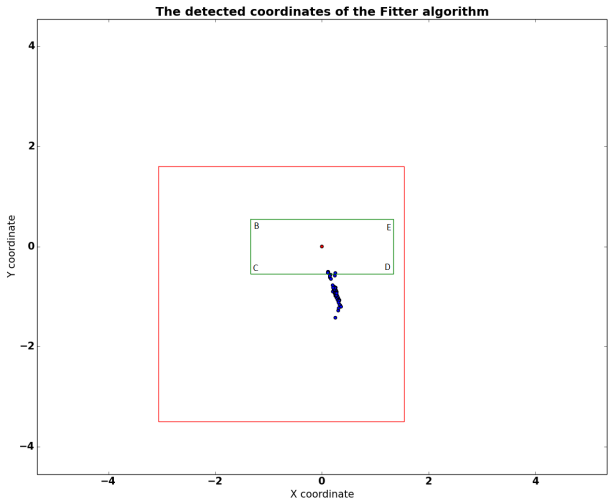


(b) The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 4(after)

Figure A.37: The states and coordinates detected by the Fitter algorithm for the iPhone mobile device in experiment 3 Test 4(before and after)



(a) The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 4(before)



(b) The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 4(after)

Figure A.38: The reconstructed coordinates by the Fitter algorithm for the iPhone mobile device experiment 3 Test 4(before and after)

Bibliography

[1] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI magazine, 17(3), 37.

[2] Shaw, M. J., Subramaniam, C., Tan, G. W., & Welge, M. E. (2001). Knowledge management and data mining for marketing. Decision support systems, 31(1), 127-137.

[3] Davenport, T. H., Harris, J. G., De Long, D. W., & Jacobson, A. L. (2001). Data to knowledge to results: building an analytic capability. California Management Review, 43(2), 117-138.

[4] Loraine Charlet, A., & Kumar, D. A. (2012). Market Basket Analysis for a Supermarket based on Frequent Itemset Mining. International Journal of Computer Science Issues (IJCSI), 9(5).

[5] Rekimoto, J., Miyaki, T., & Ishizawa, T. (2007, September). LifeTag: WiFi-based continuous location logging for life pattern analysis. In LoCA (pp. 35-49).

[6] Zhang, Z., Zhou, X., Zhang, W., Zhang, Y., Wang, G., Zhao, B. Y., & Zheng, H. (2011, September). I am the antenna: accurate outdoor AP location using smartphones. In Proceedings of the 17th annual international conference on Mobile computing and networking (pp. 109-120). ACM.

[7] Jiang, J. A., Tseng, C. L., Lu, F. M., Yang, E. C., Wu, Z. S., Chen, C. P., ... & Liao, C. S. (2008). A GSM-based remote wireless automatic monitoring system for field information: A case study for ecological monitoring of the oriental fruit fly,Bactrocera dorsalis. Computers and electronics in agriculture, 62(2), 243-259.

[8] Chon, J., & Cha, H. (2011). Lifemap: A smartphone-based context provider for location-based services. Pervasive Computing, IEEE, 10(2), 58-67.

[9] Montoliu, R., & Gatica-Perez, D. (2010, December). Discovering human places of interest from multimodal mobile phone data. In Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia (p. 12). ACM.

[10] Varshavsky, A., de Lara, E., Hightower, J., LaMarca, A., & Otsason, V. (2007). GSM indoor localization. Pervasive and Mobile Computing, 3(6), 698-720.

[11] Zhao, X., Lau, R. S. M., & Lam, K. (2002). Optimizing the service configuration with the least total cost approach. International Journal of Service Industry Management, 13(4), 348-361.

[12] Al Nuaimi, K., & Kamel, H. (2011, April). A survey of indoor positioning systems and algorithms. In Innovations in Information Technology (IIT), 2011 International Conference on (pp. 185-190). IEEE.

[13] Zeimpekis, V., Giaglis, G. M., & Lekakos, G. (2002). A taxonomy of indoor and outdoor positioning techniques for mobile location services. ACM SIGecom Exchanges, 3(4), 19-27.

[14] Woodman, O., & Harle, R. (2008, September). Pedestrian localisation for indoor environments. In Proceedings of the 10th international conference on Ubiquitous computing (pp. 114-123). ACM.

[15] Boztuğ, Y., & Reutterer, T. (2008). A combined approach for segment-specific market basket analysis. European Journal of Operational Research, 187(1), 294-312.

[16] Shaw, M. J., Subramaniam, C., Tan, G. W., & Welge, M. E. (2001). Knowledge management and data mining for marketing. Decision support systems, 31(1), 127-137.

[17] `http://en.wikipedia.org/wiki/Friis_transmission_equation`

[18] A. LaMarca et al.(2005). Place Lab: Device Positioning Using Radio Beacons in the Wild. Proc. 3rd Int'l Conf. Pervasive Computing, LNCS 3468, Springer, pp. 116–133.

[19] Prasithsangaree, P., Krishnamurthy, P., & Chrysanthis, P. K. (2002, September). On indoor position location with wireless LANs. In Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on (Vol. 2, pp. 720-724). IEEE.

[20] Bahl, P., & Padmanabhan, V. N. (2000). RADAR: An in-building RF-based user location and tracking system. In INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (Vol. 2, pp. 775-784). Ieee.

[21] Hightower, J., & Borriello, G. (2001). A survey and taxonomy of location systems for ubiquitous computing. IEEE computer, 34(8), 57-66.

[22] Cornelius, C., Kapadia, A., Kotz, D., Peebles, D., Shin, M., & Triandopoulos, N. (2008, June). Anonysense: privacy-aware people-centric sensing. In Proceedings of the 6th international conference on Mobile systems, applications, and services (pp. 211-224). ACM.

[23] Kapadia, A., Kotz, D., & Triandopoulos, N. (2009, January). Opportunistic sensing: Security challenges for the new paradigm. In Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International (pp. 1-10). IEEE.

[24] Consolvo, S., Smith, I. E., Matthews, T., LaMarca, A., Tabert, J., & Powledge, P. (2005, April). Location disclosure to social relations: why, when, & what people want to share. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 81-90). ACM.

[25] Harris, L., & Westin, A. F. (2003). Consumer Privacy Attitudes: A Major Shift Since 2000 and Why.

[26] Sadeh, N., Hong, J., Cranor, L., Fette, I., Kelley, P., Prabaker, M., & Rao, J. (2009). Understanding and capturing people's privacy policies in a mobile social networking application. Personal and Ubiquitous Computing, 13(6), 401-412.

[27] http://www.polestar.eu/en/

[28] http://www.forbes.com/sites/kashmirhill/2014/06/20/why-a-san-francisco-coffee-shop-stopped-tracking-customers-phones/

[29] http://euclidanalytics.com/

[30] http://www.purplewifi.net/

[31] http://retailnext.net/

[32] http://www.infsoft.com/

[33] Conway, J., Pseudoexperiments,Pulls,and Kinematic Fitting, Physics 252C - Lecture 15.

[34] Mautz, R. (2012). Indoor positioning technologies (Doctoral dissertation, Habil. ETH Zürich, 2012).

[35] JCGM 200:2008: International Vocabulary of Metrology Basic and General Concepts and Associated Terms, 3rd Edition, Joint Committee for Guides in Metrology.

[36] Thomas, F., & Ros, L. (2005). Revisiting trilateration for robot localization. Robotics, IEEE Transactions on, 21(1), 93-101.

[37] http://en.wikipedia.org/wiki/Trilateration (March 20th 2014)

[38] Demortier, L., & Lyons, L. (2002). Everything you always wanted to know about pulls. CDF note, 43.

[39] Chon, Y., Talipov, E., Shin, H., & Cha, H. (2011, November). Mobility prediction-based smartphone energy optimization for everyday location monitoring. In Proceedings of the 9th ACM conference on embedded networked sensor systems (pp. 82-95). ACM.

[40] Koyuncu, H., & Yang, S. H. (2010). A survey of indoor positioning and object locating systems. IJCSNS International Journal of Computer Science and Network Security, 10(5), 121-128.

[41] http://en.wikipedia.org/wiki/Root-mean-square_deviation

[42] Want, R., Hopper, A., Falcao, V.,& Gibbons, J. (1992). The active badge location system. ACM Transactions on Information Systems (TOIS), 10(1), 91-102.

[43] Sayed, A. H., Tarighat, A., & Khajehnouri, N. (2005). Network-based wireless location: challenges faced in developing techniques for accurate wireless location information. Signal Processing Magazine, IEEE, 22(4), 24-40.

[44] Pahlavan, K., Li, X., & Makela, J. P. (2002). Indoor geolocation science and technology. Communications Magazine, IEEE, 40(2), 112-118.

[45] Gu, Y., Lo, A., & Niemegeers, I. (2009). A survey of indoor positioning systems for wireless personal networks. Communications Surveys & Tutorials, IEEE, 11(1), 13-32.

[46] Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian P. (2007). "Preface to the Third Edition". Numerical Recipes: The Art of Scientific Computing (2rd ed.). New York: Cambridge University Press. p. xi.

[47] Stites D., Skinner K., User privacy on iOS and OS X

[48] http://www.techtimes.com/articles/8233/20140612/
apple-implements-random-mac-address-on-ios-8-goodbye-marketers.
htm

[49] http://www.tomsguide.com/us/ios-8-mac-address-randomization,
news-18937.html

[50] http://www.washingtonpost.com/blogs/the-switch/wp/2014/06/09/
how-apples-new-software-makes-it-harder-for-retailers-to-track-your-movements/

[51] http://thenextweb.com/apple/2013/03/26/
what-exactly-wifislam-is-and-why-apple-acquired-it/

[52] http://www.pcmag.com/article2/0,2817,2416988,00.asp

[53] http://marketingland.com/apple-killed-location-tracking-ios-8-86908

[54] https://angel.co/wifislam

[55] Amoraal, J. M. (2011). Alignment with Kalman filter fitted tracks and reconstruction of B decay.

[56] http://tweakers.net/nieuws/93547/dixons-mycom-en-icentre-volgen-klanten-door-peilen-wif
html