Delft University of Technology
Master's Thesis in Embedded Systems

Analysing TDoA Localisation in LoRa Networks

David Bissett

embedded software

TU Delft Delft University of Technology

Analysing TDoA Localisation in LoRa Networks

Master's Thesis in Embedded Systems

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

David Bissett
d.w.j.bissett@student.tudelft.nl

1st October 2018

Author
David Bissett (d.w.j.bissett@student.tudelft.nl)
Title
Analysing TDoA Localisation in LoRa Networks
MSc presentation
8th October 2018

Graduation Committee

dr.ir. F.A. Kuipers (chair)	Delft University of Technology
dr. M.A. Zúñiga	Delft University of Technology
dr. N. Yorke-Smith	Delft University of Technology

Abstract

The IoT contains billions of interconnected devices and is only growing larger by the day. These devices often need to operate for years while keeping mobile objects, such as animals or vehicles, connected to a larger system. Therefore they will be battery-powered, energy efficient and, their position will be tracked. Therefore, connecting these devices requires low-power, long-range communication that can also be used for localisation. Because of the scale of the IoT industry, manufacturing and infrastructure costs need to be considered. The low cost, long range and energy efficiency that is required exempts many wireless communication and localisation solutions such as WiFi, GPS and Bluetooth because they are not as suitable as Low-Power Wide-Area Networks (LPWAN's). LoRa is a LPWAN technology that is both Long-Range and cost effective. Therefore, the objective of this thesis is to use a LoRa network for our localisation algorithms.

In this work we show that signal strength data becomes turbulent when communicating over a large, urban area. Therefore we evaluate Time Difference of Arrival (TDoA)-based localisation algorithms, including a novel area-based algorithm that we developed. We evaluate the localisation algorithms on a proprietary LoRa network which also provides a localisation service that we use as a benchmark. We evaluate the performance of the algorithms over a large, mostly urban, region of The Netherlands. Using mobile LoRa devices, we show that for 80% of cases, our proposed algorithm has a position error less than 925m. We also show that the other localisation methods, including the proprietary localisation service, have a larger maximum position error for the same portion of cases. This work contributes a localisation algorithm that can compete against proprietary geolocation services, such as Sigfox and KPN's services, in many applications.

Preface

Localisation has been an important field of study for thousands of years, since sailors needed to find their position on the ocean. In the beginning, stars and landmarks were used for positioning, calculating angles and distances by hand. Technology has opened many possibilities in this field, with processors and wireless technologies allowing us to almost instantly calculate our position and communicate it to others. In modern times, technology has advanced even further allowing these processors and communication modules to become embedded in increasingly smaller devices. This has given birth to the IoT.

I enjoy IoT projects because of their variety in both technology and application. When I spoke to Marco Zúñiga about the thesis topics he was offering, this topic grabbed my attention not only because it was an IoT-related project but also because I could make use of KPN's LoRa network. For me this was an opportunity to test my skills in the real-world, instead of attempting to simulate reality in a lab or a small field. The advantage of this was that I could use the full range of LoRa which was not done in related works.

There are many people who have helped me during this thesis. Firstly, I would like to thank Marco for his enthusiasm, support and guidance that he provided during my thesis. Secondly, I would like to thank Fernando for finding me a position at The Things Conference and giving his insights on LoRa. Thirdly, I would like to thank Ioannis for all his help and time, without which this project would have taken far longer. I would like to thank the VLC research group for their advice and support whenever I presented my latest work. I would also like to thank my friends whom have helped me to expand my comfort zone and, who gave balance to my academic and social lives. Finally, I would like to thank my family for all their love and support which has been a cornerstone in my development as an engineer.

David Bissett

Delft, The Netherlands
1st October 2018

Contents

Chapter 1

Introduction

1.1 Motivation

These days, hardly a day goes by where we don't encounter electronics, whether it be to pay for groceries or hailing a taxi from your phone. This convenience of using electronics has created a large demand for electronic gadgets, making the Internet of Things (IoT) a large area of interest. Some have even predicted that over 20 billion devices will be connected by the year 2020 [22, 37]. An added feature for many of these devices is that they can be tracked. Although this feature is worrisome to many people, it has a multitude of applications where it proves to be quite useful. This could be tracking a rhino to prevent poachers [3], tracking a weather station floating in the ocean [8] or, just tracking your pizza while a teenager delivers it on their bike. For many of these cases the devices will run on a battery requiring a lifetime of several years, possibly in a remote location requiring nearby infrastructure. As with everything, cost can become an issue; more so when a device is being mass-produced, with manufacturers aiming between three to five euros per device [24]. Some of the present day solutions for IoT localisation utilise GPS, Wi-Fi, Bluetooth or a combination of these and other technologies. However, when looking at what is needed for many IoT applications, these solutions are often suboptimal.

GPS can be useful for localisation in remote regions as satellites communicate directly to the device, avoiding the need for nearby infrastructure. However, the power and cost associated with a GPS can be disadvantageous for an IoT device going to market [27]. In most cases, the device will also need an extra module for communication. This is not a problem for Wi-Fi and Bluetooth which can also perform well for localisation in certain environments. They are mostly used for localisation in a smaller area because the range of Wi-Fi is typically only around 100m [14], and Bluetooth even less [32]. Therefore a dense network of access points will be required for these technologies to cover a large region, greatly increasing the cost of

deployment.

Power consumption, cost and range are where LoRa has its advantage over other technologies. LoRa is a low power, long range, wireless communication technology that is also cheap to implement [24]. Due to these useful properties, LoRa has gained popularity in the world of IoT [37]. It would therefore be desirable to localise devices with the LoRa signals that are being used for communication.

1.2 Problem Statement

Much research has already been done on localisation schemes that use wireless communication. But as mentioned in our motivation, there are some disadvantages for many of these solutions. LoRa does not have the same weaknesses. Thus, we would like to build upon current wireless localisation algorithms, but instead; use LoRa technology.

To create a more general localisation solution for IoT devices it will be beneficial to create an algorithm that does not require any information from extra sensors such as accelerometers for example, as unnecessary sensors can mean unnecessary power consumption. Continuing with this theme, it is also better not to temporally synchronise the device with the receivers. This requires regular communication and thus, more energy.

In order to satisfy these constraints we have chosen to use a Time Difference of Arrival (TDoA) based approach. We formulate our main research objective to be as follows:

Develop and implement a localisation algorithm for a LoRaWAN, based on the TDoA of LoRa transmissions.

In order to test the performance of our algorithm on a LoRaWAN, our experiment needs to cover an area large enough to experience the effects of long range communication. Therefore we use a proprietary LoRa network with nationwide coverage to gather data.

1.3 Contributions

By taking inspiration from various localisation techniques for our algorithm, we have been able to observe the advantages and disadvantages of these different methods. My key contributions are:

- Evaluation of area-based techniques to limit the effect of measurement error on TDoA position estimation.
- Development of a novel TDoA-based localisation algorithm that is resistant to TDoA measurement error.

Other works on localisation of LoRa devices work in small areas, avoiding the challenging characteristics of long-range signals for positioning. In this thesis we use a nationwide LoRa network for our experiments to fully observe the difficulties associated with LoRa's long range and test how various localisation algorithms will perform with these challenges.

1.4 Plan of Development

In Chapter 2 a few related works, relevant to this thesis, will be presented. We will then give background information explaining the basics of LoRa, as well as background information for localisation techniques used in this thesis. Chapter 3 will then describe the setup that was used during experiments followed by Chapter 4 which will present the different localisation algorithms that were used and give a comparison between them. Finally we will present our conclusions in Chapter 5 along with some ideas for possible future research.

Chapter 2

Background and Related Work

This chapter will give background information regarding the topics covered in this thesis as well as presenting some works related to localisation of IoT devices. Section 2.1 will discuss some related works, followed by Section 2.2 giving information about LoRa wireless communication technology[1]. Section 2.3 discusses some localisation techniques that will be used in this thesis and finally, Section 2.4 will explain some necessary geodetic techniques.

2.1 Related Work

2.1.1 GPS Localisation

For localisation of IoT devices, the state of the art technology, in terms of accuracy, is GPS. In most modern Global Positioning System (GPS) modules, an accuracy of less than 10m can be observed in an open outdoor environment [6], with many even achieving centimetre-level accuracy [42]. Based on this, and GPS's extensive coverage, it can be quite an attractive solution for IoT localisation. On the negative side, GPS has the disadvantage of having a higher power consumption, as mentioned in Chapter 1. Using GPS for localisation consumes more than 10 times the energy of LoRa, when localisation packets are sent at the same rate. The difference can even be up to 20 times, if LoRa is configured in an efficient way [20, 34, 35]. What's more, is that in addition to a GPS module, a device will also require an extra module for communication. With this in mind, it would be more appealing to use a Low-Power Wide-Area Network (LPWAN) for both communication and localisation.

[1] Seeing as all the experiments were carried out in the Netherlands, only the EU specifications will be covered in this section.

2.1.2 Localisation Using RSSI in a LPWAN

Two technologies that can be used in a LPWAN are LoRa and, one of its competitors, called Sigfox. Sigfox is a proprietary, Ultra Narrowband (UNB), wireless communication technology owned by a company of the same name. Sigfox, like LoRa, is aimed at IoT devices thus, one of the services Sigfox offers is geolocation. The localisation algorithm that they use is trilateration with RSSI ranging, which we will briefly describe in Section 2.3. To improve their results, Sigfox are using machine learning techniques. Despite their efforts, they are still only able to locate devices with an accuracy of between 1-10km for up to 80% of cases for static devices [4]. For a higher precision (<500m), they use information from nearby Wi-Fi sources which is compared to crowd-sourced data. This requires having a Wi-Fi module in addition to using Sigfox and counteracts the advantages of using a LPWAN for both communication and localisation.

The results of Sigfox have been improved upon by a few other researchers using machine learning. One of the improved techniques was presented in [33]. In this paper, a position accuracy of less than 50m is achieved for 100% of their cases although, this figure can be misleading. In their algorithm they already know the location of the device to within a radius of 200m. They then only need to consider this small area when performing localisation with Sigfox. Considering that the range of Sigfox can be up to 40km [33], it seems that this is an underutilisation of the technology and, will not face the same difficulties associated with long range communication. This is also seen in [21], where LoRa is used for positioning, using RSSI. They achieve an accuracy of less than 20m. However, in these experiments an area of only 64x110m was considered.

Sigfox and LoRa are both capable of communication over tens of kilometres so, by only considering localisation over small areas as these works do, localisation becomes easier and the long range characteristics of these technologies are not observed. In this thesis we wish to truly test localisation algorithms, experiencing the challenges of long-range technology. Fortunately, there are a few proprietary LoRa networks with nation-wide coverage that have been established recently.

2.1.3 Proprietary LoRa Geolocation

Since the invention of LoRa, many companies have started adopting it into their repertoire of technologies. Telecommunications operators in France, Switzerland and even KPN here in The Netherlands are among those already deploying LoRa networks, adapting their existing infrastructure [12]. KPN is even offering a localisation service using their LoRa network, boasting an accuracy of less than 100m for 90% of their location estimations for stationary devices [1]. After doing a lot more research on KPN's geolocation

service, not much was discovered except that they use a localisation technique called TDoA. This revealed to us a problem with proprietary solutions as well as an opportunity for research. The problem being, that proprietary solutions remain closed source and can only be used in a black-box manner, with companies naturally withholding detailed information. Therefore, using a proprietary LoRa network, we shall develop our localisation algorithm more openly in order to benefit the scientific community. To the best of our knowledge there is not yet a localisation algorithm that has been published, that has been implemented and tested on a nation-wide LoRa network. For this thesis, we have access to KPN's geolocation service for our LoRa nodes. Thus, we will use it as a comparison for our localisation algorithm throughout this report.

Before we cover various localisation algorithms it would be best to have a basic understanding of LoRa as this will influence our choice of techniques for localisation.

2.2 LoRa Wireless Communication

LoRa wireless communication is made up of LoRa's physical layer and the LoRaWAN protocol. LoRa modulation is a patented technology that was acquired by Semtech. Therefore the amount of public documentation on LoRa modulation is limited. Nevertheless, this section will give a description of LoRa and LoRaWAN to the best of our knowledge, with a focus on features that will affect localisation.

2.2.1 LoRa

LoRa is a wireless modulation scheme, implemented at the physical layer (PHY), that can be used for Long-Range, wireless communication. The term LoRa is officially only to describe the LoRa PHY. However, in this thesis the term LoRa will be used to refer to the technology as a whole, as is commonly done in related works.

LoRa modulation is built upon Chirp Spread Spectrum (CSS) modulation, making it more robust to noise and interference. CSS works by continuously varying the frequency of the signal either up or down (up-chirp or down-chirp). This is done at a fixed rate known as the chirp rate. An illustration of an up-chirp can be seen in Figure 2.1.

The Doppler effect and multipath interference can change the frequency or timing of a signal. Because the chirp rate is fixed, there is a direct relationship between frequency and time in CSS modulation; a useful property that LoRa takes advantage of to help mitigate this kind of interference [36].

To improve upon CSS, LoRa modulation utilises various techniques to increase its resilience against noise and interference [19]. One of these methods involves encoding symbols during LoRa modulation. The number of

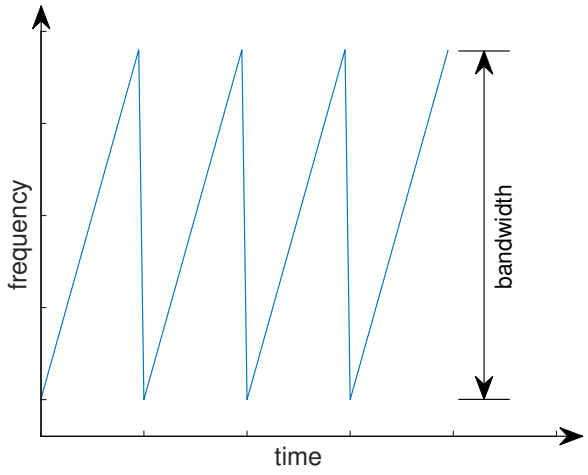


Figure 2.1: Illustration of an up-chirp in CSS modulation.

bits used to encode each symbol is equal to what is called the *Spreading Factor* (SF). The SF used by a LoRa device can be any discrete number from 7 to 12. A higher SF makes the signal more robust to noise, allowing it reach more receivers that are further away. Having more receivers improves the accuracy of localisation [21], making the SF an important parameter to consider when developing a localisation algorithm.

Using a SF of 12, LoRa can achieve a range of up to 30km in open, outdoor environments [29]. Some experiments have even found that LoRa signals can be received at distances of a few hundred kilometres, with the record standing at a distance of 702.7 km for a successfully received LoRa transmission [39]. Intuition would then conclude that a SF of 12 should always be used. However, a SF of 12 also has the slowest data rate. A slower data rate means that a device will need to be on air for a longer time to transmit the same amount of information. This has the disadvantage of creating more interference for other devices using a similar frequency, as well as creating unnecessary traffic for LoRa networks. For this reason, many network operators strongly encourage the use of an adaptive data rate (ADR) [28]. When ADR is enabled a device will use the lowest possible SF that can still achieve a stable connection [31]. This results in each LoRa transmission being received by a minimal number of gateways which can be detrimental for localisation. To understand more about the interaction between LoRa devices and the gateways it's best to look at the LoRaWAN protocol.

2.2.2 LoRaWAN Protocol

LoRaWAN is a specification, developed by the LoRa Alliance, that defines the communication protocol and network architecture used by LoRa. The

LoRaWAN protocol is implemented on top of the LoRa PHY which is agnostic to protocols implemented on top of the physical layer. LoRaWAN covers many topics including security, amongst others, but for understanding the topics presented in this thesis it is only necessary to explain the network architecture. If we look at the network topology used in a LoRaWAN, illustrated in Figure 2.2, we can get a better understanding of how LoRa devices interact with the network.

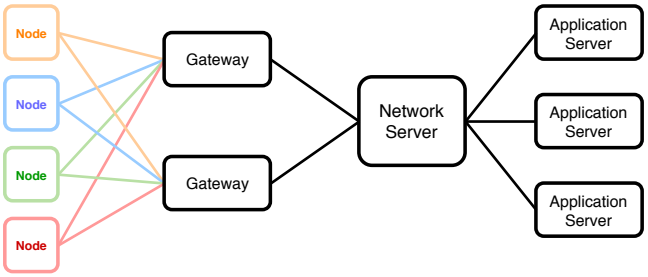


Figure 2.2: Network topology defined by LoRaWAN.

The different colours of the end nodes in this figure are used to distinguish between each individual node and, to link the nodes with their respective communication channels. Starting at the end of the network, on the left of Figure 2.2, the nodes are connected directly to the gateways via LoRa. The end nodes are not assigned a specific gateway. Rather, a transmission sent by a node is received by all the gateways that are able to receive it. When a gateway receives a transmission, it immediately gives it a timestamp as well as recording the Received Signal Strength Indicator (RSSI) and the Signal to Noise Ratio (SNR). The data from the transmission, along with the meta-data recorded by the gateway, is then sent to the network server. For localisation, this meta-data is very important, which will become clear in the next chapter. From the network server, the data is then sent to the appropriate application server. This is where the data from the end devices can be processed, for example, to perform localisation. However, in our case we download our data and process it offline.

If an application server needs to send a message to the LoRa device, the message will first be queued at the network server and then forwarded to the LoRa device during its next receive window. This is dependent on the mode of operation of the LoRa device. In [23], the LoRa Alliance lists three modes of operation for a LoRa enabled device. These modes are denoted A, B, and C and are defined as part of the LoRaWAN protocol. Mode A is the simplest mode and forms a basis for the other two modes. Therefore, when creating a generalised localisation algorithm for LoRa, only mode A needs to be considered.

Mode A is supported by all devices and, it is the most energy efficient of

the three modes. In mode A, a device only transmits a signal when it has data that it needs to send, much like the ALOHA protocol. After the device sends a signal to the network, it then opens two receive windows. Once both receive windows are closed, the server has to wait until the device's next transmission to send a message back to the device [23].

Even though this mode allows devices to be more energy efficient, it has consequences that make certain tasks more difficult. For example, in a positioning algorithm, if localisation was not possible for a certain signal, it would be difficult to request a new transmission. Another task that becomes difficult is time synchronisation. Even though this is already energy intensive for an IoT application, it becomes infeasible using mode A. Therefore, a localisation algorithm that requires a device to be synchronised with the network would not be practical. To understand a bit more about the requirements and methodology behind different localisation techniques we provide relevant explanations in the next chapter.

2.3 Localisation

Localisation is an ancient practice with early navigators using stars and landmarks to find their way thousands of years ago. These days navigation techniques remain relatively similar, replacing stars and landmarks with satellites and radio towers. Of course, there are also many new techniques that have been developed over time. This section will describe some of the basic principles of localisation as well as some fundamental techniques.

2.3.1 Localisation Basics

In order to locate a device we first need some points of reference with known locations which we call *anchor points*. Using these anchor points, there are three popular techniques that can be used to locate a device. These techniques are triangulation, trilateration and multilateration.

In triangulation, the location of a device can be estimated using the geometry of a triangle that is formed between two anchor points and the device. The geometry of the triangle can be calculated using the Angle of Arrival (AoA) of a transmission from the anchors to the device, or vice versa. However, using AoA measurements is not suitable for LoRa's long range because the errors become larger the further away a device is from the anchor points.

The next technique to consider is Trilateration, which has already been used in research on localisation for LoRa devices [21, 40]. Trilateration uses the distance between a device and each anchor point. The distance from a device can be estimated in two ways: using Time of Arrival (ToA) measurements or, RSSI ranging.

In the ToA method, the time of transmission must be known, requiring the device to be synchronised with the network and to use a precise clock. This

requires extra communication overhead and, extra cost; not ideal for cheap, battery-powered devices. Therefore, ToA will not be considered further.

RSSI ranging has already been used for trilateration in LoRa networks thus, it would be remiss not to discuss it in this report.

2.3.2 RSSI Ranging

The basic principle behind using the Received Signal Strength Indicator (RSSI) for ranging is to use the signal power that is lost between transmission and reception to calculate the distance covered by the signal. The signal power that is lost over the communication medium is known as the *path loss*. The path loss can be determined from what is called the *link budget*, a summary of all gains and losses from the transmitter, through all transmission mediums, to the receiver. This can be characterised in a simple manner with the equation [36]:

$$P_{Rx} = P_{Tx} + G_{Rx} + G_{Tx} - L_{PL} \tag{2.1}$$

where P_{Rx} is the signal power at the receiver, in dB, where we can substitute the value of RSSI. P_{Tx} is the signal power at the transmitter and, G_{Rx} and G_{Tx} are the gains of the antennae used by the receiver and transmitter respectively. Once we have substituted RSSI into (2.1) to calculate the path loss, we need to relate it to distance in order to calculate how far the signal travelled.

Theoretically, as a wireless signal propagates through free space, the path loss increases proportionally to the square of the distance it travels from the point of transmission [15]. In reality, this is not usually the case as there are many sources of interference, especially in urban areas. In this case, it is more appropriate to estimate distance using the log-distance path loss formula [10]:

$$L_{PL}(d) = L_{PL}(d_0) + 10\gamma log\left(\frac{d}{d_0}\right) + X_\sigma \tag{2.2}$$

where $L_{PL}(d)$ is the path loss, in dB, at distance d, $L_{PL}(d0)$ is the path loss at a reference distance d_0, γ is the path loss exponent and, X_σ is the loss from shadow fading with a zero mean Gaussian distribution and a standard deviation of σ.

Substituting the path loss into (2.2) and rearranging the formula, we can solve for the distance d. But first, we need to obtain values for γ and, $L_{PL}(d_0)$. This can be done using empirical measurements. In [29], Petäjäjärvi et al. found that $\gamma = 2.32$ and $L_{PL}(d_0) = 128.95dB$ in the city of Oulu, in Finland. However, in [18], Jörke et al. found that $\gamma = 2.65$ and $L_{PL}(d_0) = 132.25dB$ in the city of Dortmund, in Germany. The discrepancy in these values was due to different building densities for Oulu and Dortmund

[18]. Therefore, we cannot assume a general value for γ or $L_{PL}(d_0)$ and will have to estimate these values using our own measurements.

Works that have located LoRa devices with RSSI either used a small area [21] or, work in open, rural areas with minimal interference [3, 40]. However, obstacles in an environment severely increase the variability of RSSI measurements, relating to distance variability of up to 5km [18] or more [29]. Therefore, in Chapter 4 we will only show the variability of RSSI ranging in a large urban environment and will not be using it for localisation. A more promising technique is TDoA localisation which we will explain in the next section.

2.3.3 TDoA

As mentioned in Section 2.3.1, there is a third common method of localisation called multilateration. This method does not require the exact distance from a device to each anchor point but rather, only the differences in distance from each gateway to the device. The difference in distances can be calculated with the Time Difference of Arrival (TDoA) of a signal from a device to the anchor points or vice versa. Because of this, it is often referred to as TDoA localisation.

TDoA is a popular technique for localisation as it does not require the transmitter to be synchronised with the receivers. This is because TDoA only requires the differences between the timestamps of a transmission. If we recall from Section 2.2.2, these timestamps are given at each gateway when receiving a transmission. Because TDoA does not require a timestamp from the device, the receivers only need to be synchronised with each other. An explanation of how to use TDoA for localisation will be given below.

Let us say that when a LoRa signal is transmitted from a device, it is received by n gateways. These gateways will be our anchor points because we know their locations. Each gateway will be at a slightly different distance to the device therefore, they will receive the LoRa transmission at different instances in time.

Because TDoA uses the difference in time, there is one measurement for each possible pair of gateways. The total number of possible pairs, for a certain transmission, is then given by the binomial coefficient: $\binom{n}{2}$. For each of these gateway pairs, the Time Difference of Arrival (TDoA) can be represented by $\Delta t_{i,j} = t_j - t_i$, where $1 \leq i < j \leq n$ and, t_i and t_j are the timestamps for gateways i and j.

Theoretically, by using the time differences from all the possible gateway pairs, we can then calculate the position of the transmitter if the signal was received by at least three gateways. The differences in the two timestamps for a gateway pair, $\Delta t_{i,j}$, shall be referred to as the TDoA measurement. The distance that can be calculated from the TDoA measurement will be referred to as the TDoA distance. This distance can be calculated using the

common velocity formula and solving $\Delta d_{i,j} = c\,\Delta t_{i,j}$ where c is the speed of light through air.

Using a TDoA measurement, we can create a hyperbola consisting of all the possible points of where the node could be, i.e. points with a constant TDoA measurement for a given gateway pair. Gustafsson and Gunnarsson give a derivation of the equation for such a hyperbola in [16] which will be explained in this section.

To compute the hyperbola, we have to assume that the two gateways in a pair are situated on the x-axis, equidistant from the y-axis. Let us define the distance between the two gateways as D. Therefore the gateways lie at $x = D/2$ and $x = -D/2$. The hyperbola can then be represented by all the possible points (x, y) calculated with the equation shown below.

$$\frac{x^2}{\Delta d^2/4} - \frac{y^2}{D^2/4 - \Delta d^2/4} = 1 \tag{2.3}$$

Where Δd is the difference in distance calculated from the respective TDoA measurement.

To calculate the asymptote of the hyperbola, we solve for y, in (2.3), for when $x \gg \Delta d^2/4$ and $y \gg D^2/4 - \Delta d^2/4$:

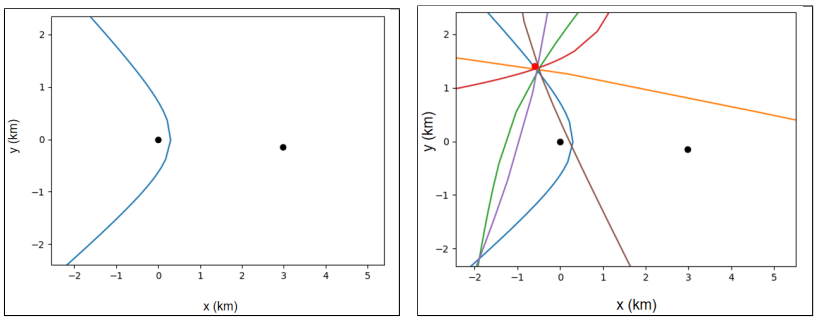
$$y = \pm\sqrt{\frac{D^2/4 - \Delta d^2/4}{\Delta d^2/4}}x \tag{2.4}$$

The derivation of (2.3) and (2.4) from [16] has been added to Appendix A.

It should be noted that the $\begin{pmatrix} x \\ y \end{pmatrix}$ coordinates used for drawing a hyperbola are only "*local coordinates*" that are specific to a pair of gateways. This means that they need to be mapped onto the 2D representation of the Earth, called the "*global coordinates*" [16]. Only then can each hyperbola be related to one another. Converting from local coordinates to global coordinates simply involves a rotation and then a translation of the two-dimensional plane. This is expressed as follows [16]:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix} + \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha & cos(\alpha) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{2.5}$$

where $\begin{pmatrix} x \\ y \end{pmatrix}$ are the local coordinates to be converted, $\begin{pmatrix} X \\ Y \end{pmatrix}$ are the resulting global coordinates and, $\begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$ is the midpoint between the two gateways. Lastly, $\alpha = arctan\left(\frac{Y_i - Y_j}{X_i - X_j}\right)$ represents the angle formed between the two gateways and the global x-axis. Using (2.5), we can represent the hyperbolic function of a TDoA measurement in global coordinates. An example is shown in Figure 2.3(a).



(a) (b)

Figure 2.3: Hyperbola representing all the possible node positions for a single, ideal, TDoA measurement (left). Hyperbolae from multiple, ideal, gateway pairs' TDoA measurements intersecting at the node's position (right).

In Figure 2.3 the black dots represent LoRa gateways, and the red dot represents the node's actual position, in global coordinates. On the right-hand side we can see how the TDoA measurements from other gateway pairs (not shown) generate more hyperbolae. As we can see in Figure 2.3(b), the hyperbolae intersect at different points but the single point where all hyperbolae intersect each other indicates the node's position. This example of TDoA describes an ideal situation but in reality error arises from TDoA measurement uncertainty [16]. In Chapter 4 we will cover this in more detail.

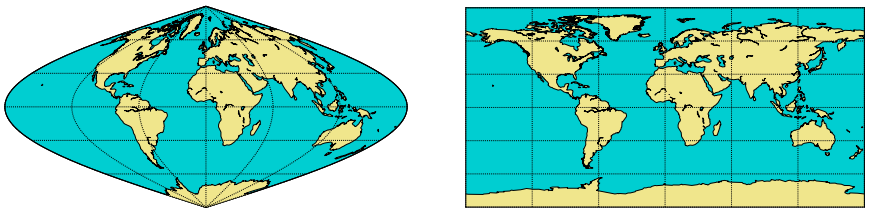
2.4 Geodesy for Localisation

In order to implement the localisation algorithms from Section 2.3, it's easiest to work in two dimensions. Therefore in Section 2.4.1 we describe a method of map projection which is used to represent the Earth's surface as a two-dimensional (2D) plane. We will then present a technique in Section 2.4.2 to find the distance between two points on the Earth, which can be used to validate the 2D coordinate system that is used.

2.4.1 Equidistant Cylindrical Projection

To represent the Earth's surface as a 2D plane, a map projection technique needs to be used. Because the Earth is an ellipsoid and not flat, projecting the Earth's surface onto a 2D plane introduces distortions. Each method of projection typically tries to preserve certain geographical features. Simultaneously, other features become distorted. We present two examples in

Figure 2.4.[2]



(a) (b)

Figure 2.4: Illustration of sinusoidal projection (left) and equidistant cylindrical projection (right).

On the left of this figure we see sinusoidal projection which preserves area and, also preserves distances along the lines of latitude. However, it distorts the lines of longitude [38]. On the right we see an example of cylindrical projection. In this thesis, we will only consider cylindrical projections that use the Earth's poles as bounds for the y-axis in the 2D plane, as shown on the right of Figure 2.4. This is because, in this type of projection, the lines of latitude and longitude remain parallel with the x and y axes respectively. Therefore, we will not have to compensate for any curve in the axes when mapping out various geographical positions or plotting hyperbolae.

The most common cylindrical projection is Mercator projection, which is used in most school text books and is even used by Google Maps [2, 38]. Mercator projection not only stretches lines of latitude to keep them the same length but, it also stretches the lines of longitude to preserve angles for navigation. The distortion that this introduces would complicate localisation and, its projection formulae are more computationally expensive than other cylindrical projections. Therefore, we will not be considering it further.

A much simpler method is the Equidistant Cylindrical projection, shown in Figure 2.4. This technique was credited to Marinus of Tyre circa A.D. 100. Almost 2000 years later, it is still used by a few well known organisations such as NASA and the US Geological Survey [26, 38]. This technique projects lines of latitude and longitude as equidistant, straight lines, resembling a Cartesian plane and, preserving distances along the lines of longitude. The equations that are used to convert a location's geodetic coordinates to Cartesian coordinates (i.e. latitude and longitude to x and y) are as follows [38]:

$$x = R\,(\lambda - \lambda_0)\,cos\,\phi_1 \tag{2.6}$$

$$y = R\,(\phi - \phi_1) \tag{2.7}$$

[2]The Basemap toolkit from Matplotlib for Python was used to draw these maps. The data outlining land masses are from openstreetmap and were downloaded from [25]

where R is the radius of the Earth and, ϕ and λ are the latitude and longitude respectively. λ_0 is known as the central meridian, which is the line of longitude that will be used as the y-axis. ϕ_1 is known as the standard parallel, which is the line of latitude that will be used as the x-axis. The standard parallel is used as the reference distance thus, it is true to scale and has no distortions along its length [38]. All other lines of latitude are projected to the same length, stretching if nearer the poles and shrinking if nearer the equator. Therefore, the standard parallel needs to be chosen near the centre of the region being considered for localisation to minimise distortion.

Once we estimate the x and y coordinates of a device, we will need to convert back to latitude and longitude. This can be done using the inverse equations of (2.6) and (2.7):

$$\phi = \frac{y}{R} + \phi_1 \tag{2.8}$$

$$\lambda = \lambda_0 + \frac{x}{R\,cos\,\phi_1} \tag{2.9}$$

To quantify the position error of our algorithms we can calculate the distance between our estimation and the actual position of a device. When working in a 2D plane it is easiest to use the Pythagorean distance formula. However, the distortions of map projection may cause error in our experiments which should be evaluated to ensure that it is within acceptable limits. Therefore, we need a technique to find the distance between two geodetic coordinates.

2.4.2 Great-Circle Distance

A technique often used in surveying to find the distance between two geographical locations, is the great-circle distance [38]. In this technique, the Earth is assumed to be a sphere with a constant radius to simplify calculations. In reality the Earth is an ellipsoid, therefore this assumption will introduce some error in the distance calculation but, this is never more than 0.5% of the actual distance [5]. To calculate the Great-Circle distance we use the formula shown below [41].

$$\Delta d = R\,cos^{-1}(sin(\phi_1)sin(\phi_2) + cos(\phi_1)cos(\phi_2)cos(\lambda_1 - \lambda_2)) \tag{2.10}$$

where Δd is the distance between two points on a sphere, R is the radius of the Earth and, ϕ_1, λ_1 and ϕ_2, λ_2 represent the latitude and longitude of the first and second point respectively.

This formula can be used for validating our position error calculations but before that is possible, we need to collect data and perform our experiments. The setup that we use to do this will be described in the next chapter.

Chapter 3

System and Experimental Setup

In this section we will describe the setup and method that we used to gather data. In Section 3.1 we will give details about the nodes that were used to gather data. Section 3.2 will then describe the LoRa network that was used, which will be followed by Section 3.3 where we will explain how and where our data was collected and how it was filtered. Finally, Section 3.4 will describe our software framework for implementing the localisation algorithms.

3.1 Mobile LoRa Nodes

As we have already mentioned, we want to test the performance of our localisation algorithm over a large region. Within this region, we also need data from many different locations. The only feasible way to achieve this would be to use mobile LoRa nodes, carried by volunteers to many different locations within The Netherlands. Fortunately, our research group already has LoRa capable devices that can be used, and are small enough to fit in the someone's pocket. The devices that we are using are the TraceME TM-901 / N1C2 from KCS shown in Figure 3.1.

It should be noted that the nodes are proprietary, limiting the configuration options. The nodes send a transmission every 5 minuted when they detect they are moving and every 15 minutes when stationary. These intervals could not be changed for our devices. The LoRa module used by the N1C2 is the SX1272 transceiver module from Semtech. According to the N1C2 datasheet [20], it has a receive sensitivity of -137dBm. For our experiments we use the maximum transmission power allowed in the subbands that we use which is 14dBm. The frequency that we use for LoRa communication is 868MHz. The nodes are set to use mode A because this mode is supported by all LoRa devices and is the most energy efficient, as we mentioned in Section 2.2.2.

Figure 3.1: One of the mobile LoRa nodes that were carried by volunteers to gather data.

The payload from the mobile nodes includes the GPS location of the node, at the time of transmission. This serves as the ground-truth for validating our localisation results. The GPS module on-board the N1C2 is the Quectel L70 GPS module. This module has an accuracy of 2.5m CEP (Circular Error Probability), meaning that 50% of GPS measurements are within 2.5m of the device's actual location. Once the transmission is sent from the LoRa device, it is received by multiple gateways. It is then handled by the LoRa network that we use, perhaps the most important part of our system.

3.2 LoRa Network

The most critical property of a network for localisation is the accuracy of the synchronisation of gateways This is because it directly impacts the accuracy of the system. An error of 1μs in a TDoA measurement relates to 300m error. Seeing as we need such accurate synchronisation and we would like to cover a large area, the best option is to use KPN's LoRa network which has timestamps on the order of nanoseconds and nation-wide coverage. To collect the data, we usedKPN's ThingPark® which not only provides data, but also the meta-data that we need for localisation. For each transmission that was sent by our nodes, the wireless logger stores the timestamps from each gateway that receives it (up to nine decimal places, and up to 10 gateways), the Signal-to-Noise Ratio (SNR), the Received Signal Strength Indicator (RSSI) and, what KPN calls the Estimated Signal Power (ESP) which will be explained in Section 4.1. The wireless logger also included the

ID's of the gateways that received each transmission which could then be used to look up the gateway locations.

3.3 Data

This section will describe the method that we use to collect our data, as well as how we select the data that is suitable for localisation.

3.3.1 Collection

As we have already mentioned, we need multiple nodes gathering data from many locations. Therefore, we use 19 N1C2 nodes to collect data using volunteers. Overall we collected the data and meta-data of just over 20 thousand LoRa transmissions. This was mostly collected in the cities of Delft and Amsterdam. However, when travelling around the Netherlands, volunteers took a node with them. The GPS locations of all the data points have been plotted in red in Figure 3.2.

The railways of the Netherlands have been plotted in grey and we can see that a few node trails seem to follow these tracks[1]. Besides these uplinks from trains, most of the volunteers' data are from when they were cycling, walking, or stationary.

3.3.2 Data Selection

Once we have downloaded the data from KPN's wireless logger, we need to filter out data that cannot be used for localisation. Filtering is done in three rounds, or levels, as we will refer to them.

For level 1 filtering, recall that the data for each transmission contains the meta-data for each gateway that received it. Level 1 filtering removes gateways from a transmission's meta-data if the gateway did not properly record a timestamp or, if the gateway's location is unknown. For example, if a transmission was received by nine gateways and we did not know the position of one of the gateways, we would remove its meta-data from the transmission. It would then appear as if this transmission was only received by eight gateways. These remaining gateways will be used as anchor points in our localisation algorithms. After level 1, we can perform level 2 filtering. This level simply removes all transmissions that have fewer than three anchor points.

Level 3 filtering is performed whenever TDoA measurements are used. The need for this stems from the mathematical limits imposed by (2.3). To obey these limits, we need to reject any TDoA distances that are larger

[1]The Basemap toolkit from Matplotlib for Python was used to draw this map. The data for railways, buildings, roads and waterways are from openstreetmap and were downloaded from [25]

Figure 3.2: The GPS locations of all the LoRa uplinks during our experiments.

than the distance between the corresponding gateways. If we think about the limits in terms of physics the maximum TDoA distance possible is the distance between the gateway pair. If the TDoA distance is greater than this it means that there is error in the timestamp, likely caused by multipath, which is undesirable. To understand level 3 filtering, let's regard a transmission as being a dataset of TDoA measurements. In this way we filter out individual TDoA measurements that violate the mathematical limits, without removing the meta-data for the corresponding anchor points. For the final stage of level 3, if there are any transmissions with fewer than three TDoA measurements, these transmissions will also be removed.

Level 1 filtering is done for every application but, on its own, is only suitable for data analysis. Level 2 is performed for all localisation algorithms because all of the algorithms that we test require at least three anchor points. Level 3 filtering is done for all algorithms that require TDoA measurements. Because our data filtering fits in these standard levels, they will form a part

of our basic software framework, which is explained in the next section.

3.4 Software Framework

We use a software framework for the localisation implementations because of the functions that they have in common. All of our data from KPN's wireless logger are downloaded as a CSV file and then processed off-line using Python.

3.4.1 Data Mapping

For our localisation algorithms, we need to be able to plot our data in a 2D plane using Global Coordinates. Therefore, we first create functions for parsing our data, using the pandas library. We then create functions to implement Equidistant Cylindrical projection for mapping coordinates to 2D. Two important input parameters for the projection functions are the standard parallel and the central meridian as they will form the origin in the 2D Cartesian plane. To improve accuracy these parameters were different for each transmission. The values that we inputted for the standard parallel and the central meridian was the centroid position of the gateways that received the transmission.

The projection functions convert individual coordinates but, area-based localisation techniques require mapping various regions. For this we used OpenCV to represent an area as a 2D array. This was a useful way of manipulating data as the array's could be viewed as grey-scale images to observe how an algorithm works. Each pixel of an image represented a 20x20m area. Therefore the size of the image was dependent on the locations of the gateways that received the transmission. We chose to vary the size of the image as opposed to varying the area represented by a pixel because LoRa's long range means that the area could be over 150km in length or less than 5km. Therefore, if a standard image size was used, the accuracy would either be compromised for larger regions or, unnecessarily accurate for smaller regions which would drastically increase the algorithms' computation time.

The last functions that we need before we can implement our algorithms are the data selection functions to filter our data. These functions implement the different filtering levels from Section 3.3.2. However, this does not conclude our framework as we still have to handle the results.

3.4.2 Validation of Results

The final piece of our software framework is data validation. To do this, we need a metric for the position error. The simplest method is using the Pythagorean distance formula on our 2D plane. However, we need to be sure

that there is not too much error introduced by map projection. Since KPN's network covers the whole of The Netherlands we will measure the distance between (50.7504°N, 3.3580°E) and (53.5550°N, 7.2278°E), the intersections of the most extreme lines of latitude and longitude in The Netherlands[2]. Using great circle distance we calculated the distance to be 408.093km. For our map projection, we used the mean latitude and longitude of the two coordinates for the standard parallel and the central meridian respectively. The distance we calculated using the Pythagorean equation is 408.215km. Therefore the most extreme error that we can expect is only 122m, or 0.02%, which is negligible[3]. Seeing as the error is so small, we will calculate position error using the Pythagorean equation on the 2D plane.

This concludes our setup for collecting and manipulating data. Of course, any system is not without its limitations. Therefore we need to examine all the possible limitations that our setup could have, and how this could affect our experiments and results.

3.5 Limitations

Due to our method of data collection, there are a few limitations that we expect may influence the results of our experiments and, the parameters of our algorithms. The limitations that we foresee are as follows:

- **Limited rural data** — Because Delft and Amsterdam are built-up cities, most of the data collected are from urban areas. Therefore we did not have much data from rural areas, which generally allow for longer range and have fewer sources of interference [12].
- **Few Modes of Transport** — Most of the volunteers use bicycles as a mode of transport or, they go by foot. This means that parameters for our algorithms or, motion-models that are inferred from this data, may not be suitable for other applications.
- **Long Intervals Between Uplinks** — Because of the 1% duty cycle rule, there are long intervals between uplinks. This is a problem inherent to most of the sub-bands used by LoRa, which means that it will not likely be suitable for applications that require frequent, real-time, location information. It also means that our algorithm may not perform as well with other wireless technologies.
- **No Height Information** — We do not have any height information for our gateways and therefore our data is forced to be in a two-dimensional plane. Adding to this, The Netherlands is very flat which

[2]These extreme lines of latitude and longitude were found using Google Maps.

[3]As we mentioned in Section 2.4.1, the standard parallel should always be near the middle of the region being considered for localisation otherwise the error can increase to 1.3%.

might be fortunate when working in two-dimensions, but may also mean that our results favour our final algorithm which may perform worse in a more mountainous region or when using aerial devices.

- **Black-box Network** — Using a proprietary network, although it has many advantages, also has many disadvantages. First, although KPN has provided us with a document containing gateway locations, not all gateway locations are listed. Second, although KPN says their network is synchronised on a nanosecond level, we do not know if the timestamps are indeed accurate to 1ns as we do not have any specifications from KPN about their time synchronisation such as jitter and latency. Therefore, the black-box nature of the network puts us at a disadvantage due to lack of information and no way of troubleshooting errors within the system.

- **ADR Enabled** — One of the configuration limits of our nodes is that we are not able to disable ADR. Disabling ADR is usually recommended by network providers for moving devices as it makes it difficult for the network to assign the optimal SF [28]. If we were able to set the SF to stay at 12 our LoRa transmissions would be received by more gateways therefore improving our localisation results [21].

Chapter 4

Localisation Implementations and Comparison

This chapter covers the development of our localisation algorithm. We have developed our final algorithm in a series of stages. At each stage we determine the results of an implementation and then try to improve on it.

To begin with, we will determine the reliability of RSSI ranging in Section 4.1. This will be followed by Section 4.2, which describes a technique that we developed for estimating an initial position, used as a start for other algorithms. The first being the TDoA NLMS localisation technique from [16] described in Section 4.3. The second being a modified version of the APIT algorithm described in. Section 4.4. Next, in Section 4.5, we will describe a novel localisation algorithm that we developed, called Multilateral Dissection (MLD). In Section 4.6 we will evaluate the effectiveness of using a device's previous position in MLD localisation, followed by Section 4.7 in which we refine the area used by MLD, by using the asymptotes of the TDoA hyperbola. Lastly, in Section 4.8, we give a summary of the chapter and a comparison of all the localisation algorithms.

4.1 RSSI Ranging

LoRa localisation has only been attempted using RSSI ranging for trilateration as far as we know, besides KPN's service. However, as we mentioned in Section 2.1, the localisation that was performed in [21] was only done over a small area. Therefore, in this section we wish to investigate the reliability of RSSI ranging in a LoRa network over a much larger urban area.

A feature that peaks our interest when considering the received power of LoRa signals is the *Estimated Signal Power* (ESP). ESP is a value that KPN adds to the meta-data of a signal but we find almost no mention of it in LoRa research, save for a few exceptions [30, 31]. Reading through these exceptions we learnt that ESP is quite useful for RSSI ranging in LoRa

networks. Because of the modulation that LoRa uses, LoRa signals are able to be received below the noise floor. ESP caters for this as it represents the RSSI value with noise taken into account. Although we can simply use the ESP values that KPN provides, we will show how ESP relates to RSSI to provide a deeper understanding.

First, let's express RSSI in terms of the received signal power and the level of noise, shown below [30]:

$$RSSI_{(Watt)} = P_{Rx(Watt)} + N_{Watt} \tag{4.1}$$

If we then replace $N_{(Watt)}$ in terms of $SNR_{(Watt)} = \frac{P_{Rx(Watt)}}{N_{(Watt)}}$ we can solve for $P_{Rx(Watt)}$ which is the ESP in this case. In logarithmic form, we are left with the following formula [30]:

$$ESP_{(dBm)} = RSSI_{dBm} + SNR_{(dB)} - 10log_{10}\left(1 + 10^{0.1SNR_{(dB)}}\right) \tag{4.2}$$

Note that if ESP values were not given to us by KPN, we could calculate them in this way. Once we know the ESP we can then continue with RSSI ranging in the same manner as we described in Section 2.3.2; use the ESP as the received signal power, P_{Rx}, in (2.1) to calculate the path loss, then calculate distance using (2.2).

If we recall from Section 2.3.2, we need to find the path loss exponent γ and the reference path loss $L_{PL}(d_0)$. To estimate these values, we need to measure the path loss at different distances. To measure this we used our dataset with level 1 filtering. However, this filtering is not enough for analysing path loss. To control as many variables as possible, we chose a single gateway in Delft that had received a large portion of LoRa transmissions. If we recall from Section 2.2.1, a higher SF enables a transmission to be received at further distances. Therefore we only selected transmissions that used a SF of 12, for which we had the most data. This left us with 3620 path loss values and corresponding distances. These data points can be seen in Figure 4.1.

From this plot we can see that most of the data points are within 15km. However, there are also a few data points that exceed 100km for the same path loss. This tells us that most of this data was affected by interference, most likely from buildings blocking the signals' paths. We fitted a logarithmic line of best fit to our data in order to determine the parameters for the path loss equation. From the line of best fit, we found $\gamma = 1.25$ and $L_{PL}(d_0) = 155.03dB$. However, this data had a weak correlation, with $r^2 = 0.1149$. This was expected as our experiment was performed in an urban area. However, this means that we cannot trust the path loss parameters that we calculated. When we repeated this experiment for other gateways, we were faced with the same problem, weak correlation. Therefore

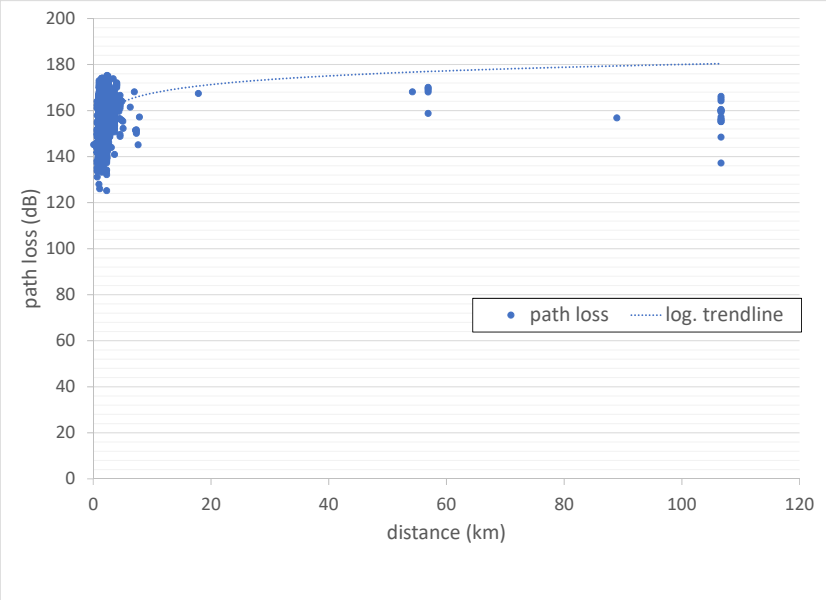


Figure 4.1: Plot of path loss (dB) vs. distance (km)

we will not pursue trilateration with RSSI ranging as a localisation solution. Rather, we shall focus on techniques that use TDoA measurements.

4.2 Initial Position Estimate

Some of the algorithms in this chapter require an Initial Position Estimate (IPE). Therefore, we will describe our method for estimation here to avoid repetition.

For our IPE we do not need it to be extremely accurate as this will be taken care of by other algorithms. The options for a rough position estimate would be to use either a range-based localisation method, or a range-free method. Our options for range-based techniques either involve time, which is what we are currently attempting already with TDoA, or RSSI which is unreliable in an urban environment as we've shown in the previous section.

Therefore, we investigated range-free localisation techniques. In [11] a centroid-based algorithm is proposed. In this algorithm all the anchor points send out a beacon containing their respective locations. The node can then estimate its position using the following formula which calculates the mean

x and y coordinates from all the received anchor positions:

$$\hat{x}, \hat{y} = \frac{x_1 + ... + x_n}{n}, \frac{y_1 + ... + y_n}{n} \tag{4.3}$$

where $\hat{x}, \hat{y}$ represent the position estimate and, n represents the number of gateways.

Using only the mean position for an estimation would not be an optimal solution for LoRa, as its long range can result in a very distant gateway still receiving the transmission and, having a large influence on the centroid. In [9], Blumenthal et al. use a weighted centroid algorithm. Their weights were based on the Link Quality Information provided by Zigbee, which is not available in LoRa communication. Therefore, we devise our own weighting system.

With RSSI measurements being unreliable, we chose to use the timestamps at each gateway to determine its weight on the centroid. Intuitively, the earlier the timestamp, the closer the gateway should be to the node. Using this intuition we can create a formula for weighting the position of gateway i:

$$w_i = \frac{t_n - t_i}{t_n} \tag{4.4}$$

where t_i is the timestamp of a transmission received at gateway i and t_n is the timestamp of the last gateway to receive the transmission.

A problem with this formula is that $w_i \longrightarrow 0$ when $t_i \longrightarrow t_n$. Therefore an extra term is needed to give a minimum weight for each gateway. We would like this term to be inversely proportional to n because if there are few anchor points, each gateway's position becomes more important. The resulting weight formula is then:

$$w_i = \frac{t_n - t_i}{t_n} + \frac{c}{n} \tag{4.5}$$

where c is a constant. Using this formula, we can sum all of the weighted gateway positions to get $\hat{x}, \hat{y}$, provided that we first normalise the weights so that $\sum_{i=1}^{n} \widetilde{w}_i = 1$, where $\widetilde{w}_i$ represents the normalised weights. Finally, we have the following formula for an initial position estimate:

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \sum_{i=1}^{n} \widetilde{w}_i \begin{pmatrix} x_i \\ y_i \end{pmatrix} \tag{4.6}$$

We then tested the performance of this initial estimation, using our data-set with level two filtering, leaving us with 7585 transmissions. When testing the performance of this method, we were able to empirically determine the value for the constant from (4.5) as $c = 0.1$. The median accuracy of this algorithm is 1.24km with 80% of cases being within 2.76km of the node's

actual position. Considering the simplicity of this algorithm and, the fact that Sigfox's geolocation has an error between 1-10km for 80% of cases, we can conclude that our algorithm performs rather well as an initial estimate. A downfall of this method is that it will always fall within the perimeter outlined by the anchor points because it is a centroid. Therefore it will incur a higher error when a transmission is only received by gateways on one side of the node. However, this algorithm is just an initial step which we will build upon in the sections to come.

4.3 TDoA with Normalized Least Mean Squares

Using TDoA measurements is a popular technique for localisation using wireless signals. However, as we mentioned in Section 2.3.3, TDoA measurement error is detrimental to the performance of localisation. To account for errors in the recorded timestamps, we decided to implement the Non-linear Least Mean Squares (NLMS) algorithm with Stochastic Gradient Descent (SGD) which was proposed by Gustafsson and Gunnarsson in [16].

To explain their method, let's first declare some definitions. If a node at position (X, Y), in global coordinates, sends a transmission that is received by n gateways, then for each unique pair of gateways, at positions (X_i, Y_i) and (X_j, Y_j), there is an ideal TDoA distance, $h(X, Y; X_i, Y_i, X_j, Y_j)$.

We can then define a non-linear system of equations;

$$\Delta d_{i,j} = h(X, Y; X_i, Y_i, X_j, Y_j), \qquad 1 \leq i < j \leq n \tag{4.7}$$

where $\Delta d_{i,j}$ is the TDoA distance computed from the transmission's timestamps at gateways i and j.

From this, the non-linear least squares estimate of (X, Y) is given as [16]:

$$(\hat{X}, \hat{Y}) = arg \min_{(X,Y)} \sum_{i<j} (\Delta d_{i,j} - h(X, Y; X_i, Y_i, X_j, Y_j))^2 \tag{4.8}$$

Gustafsson and Gunnarsson's preferred approach to solve this problem was to use SGD with a normalised LMS step size [16]. Ideally, this approach should iteratively estimate positions closer and closer to the node's location until it has found a position where the LMS error is at a minimum. To represent position, let $P = (X, Y)$ for simplicity's sake. A formula can then be written that will iteratively minimise the error from (4.8) [16].

$$P_{m+1} = P_m - \mu_m \, h'_P(P_m) \, (\Delta d - h(P_m)) \tag{4.9}$$

where μ_m is the normalised LMS step size. In this equation $\Delta d = (\Delta d_{1,2}, \ldots, \Delta d_{n-1,n})^T$. Likewise, $h(P)$ is related to this, following from (4.7). Lastly, $h'_P(P)$ is the derivative of $h(P)$. For the first position, when $m = 0$, we used our initial position estimate from (4.6).

The formula for the normalised LMS step size, μ_m, taken from [16] is as follows:

$$\mu_m = \frac{\mu}{(h'_P(P_m))^T h'(P_m)} \tag{4.10}$$

where μ is the LMS step size. When choosing a step-size, there is a trade-off between computation time and error. The smaller the step-size, the smaller the final estimation error will be but, the longer it will take to converge on a final estimation [7]. We heuristically determined that $\mu = 0.01$ gave us the best trade-off between speed and accuracy for localisation.

To evaluate this algorithm we used our dataset with level three filtering, leaving us with 6790 transmissions which we could use for testing. From the results it appears that this algorithm performed rather poorly, worse in fact than our initial estimate algorithm. The median position error of this algorithm was 3.16km with 80% of cases being within 17.88km of the device's actual location. After further investigation we found that a large portion of the transmissions had TDoA measurement errors that caused (4.9) to iteratively estimate positions further and further away from the device, finally converging at a distant, local, minimum error. An example can be seen in Figure 4.2.

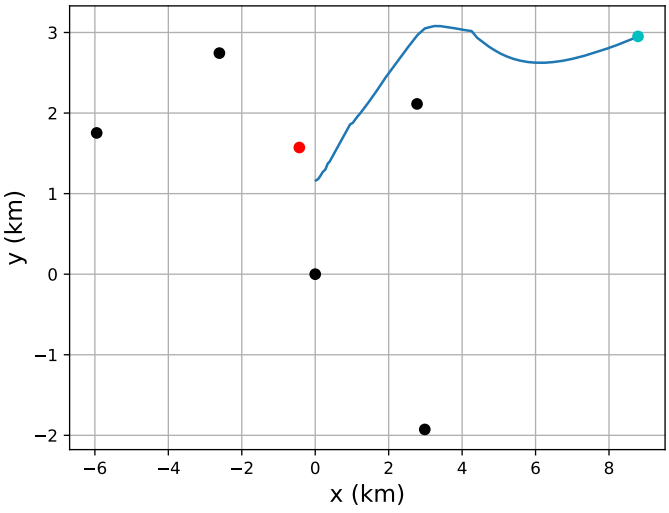


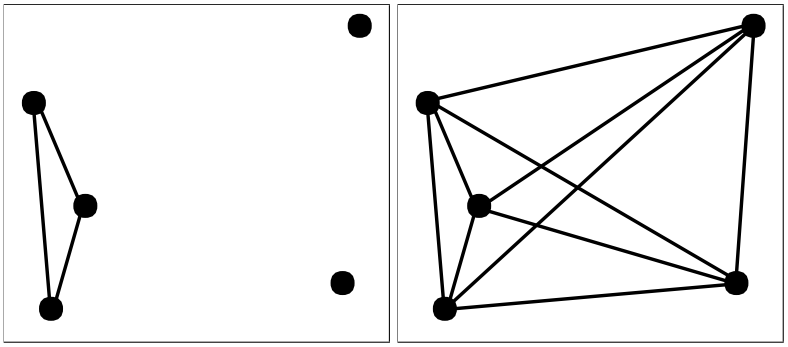
Figure 4.2: A plot showing the trace of estimations from SGD moving away from the node's location and finding a distant local minimum error.

The poor performance of this algorithm did not reflect the results from [16] which found the node's true position most of the time. However, the experiments done by Gustafsson and Gunnarsson were done in simulation

and using a zero-mean Gaussian distribution to simulate measurement error. In reality, the device will not have line-of-sight with the gateways for most transmissions. Therefore only reflections from multipath will be received at the gateways, producing errors that are not zero-mean Gaussian. Without a zero-mean distribution, (4.8) will result in large errors [13]. Therefore, this algorithm is not suitable on its own and we need a way of containing SGD to keep estimations within a certain area. Thus, we investigated an area-based localisation approach which we will describe in the next section.

4.4 Modified APIT

Here we present an area-based localisation algorithm and then explain how we modify it to suit our application. In [17], He et al. propose an algorithm for finding a region where a node is likely to reside, called APIT. For their research, APIT was developed to solve the problem of node localisation within a dense Wireless Sensor Network (WSN). In their WSN there are also many anchor nodes. To obtain the most likely area, triangles are drawn between each possible anchor trio. This is illustrated in Figure 4.3. The possibility of the node being within each triangle is then tested.



(a) (b)

Figure 4.3: Start forming triangles between 3 anchors (left) and repeat for all possible trios of anchor nodes (right).

In order to check if a node is within a triangle, He et al. proposed the Point-In-Triangulation (PIT) test. To explain the PIT test, consider three gateways, A, B and, C. The test determines if a node M lies within ΔABC, where Δ in this case indicates a triangle. The result is based on two premises:

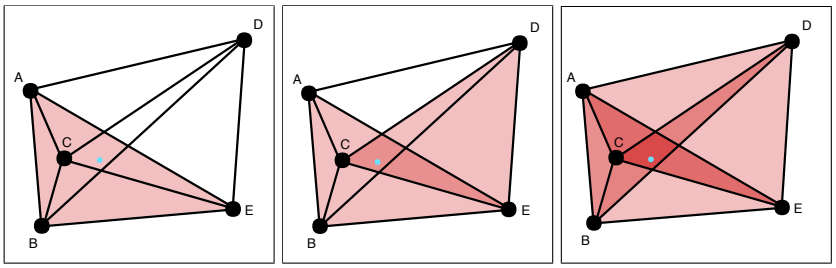
Premise 1 considers when M is within ΔABC. It states that if M is shifted in any direction, its new position will be closer to one of the anchors but further from the other two, or vice versa. This is provided that M remains within ΔABC.

Premise 2 considers when M is outside of ΔABC. If M is to be shifted, there must exist a direction in which its new position will be either further from or closer to all three anchor points. This is provided that its new position will still be outside of ΔABC.

He et al. prove these premises mathematically but, they found it was not practically feasible. Therefore, they devised an Approximate PIT (APIT) test. In the APIT test, He et al. exploit the high density of their WSN using peer-to-peer communication. They define the "***Approximate P.I.T Test****: If no neighbour of M is further from/closer to all three anchors A, B, and C simultaneously, M assumes that it is inside triangle ΔABC. Otherwise, M assumes it resides outside this triangle.*" [17].

In standard LoRa networks, peer-to-peer communication is not possible and, we do not have a dense deployment of nodes. Therefore, we need to invent our own APIT test. For our APIT test, we simply check if the IPE from Section 4.2 falls within the triangle.

In [17], for each triangle, a positive or negative weight is added depending on if the test is passed or failed respectively. In our implementation, we only add a weight if the test is passed, i.e. it contains the IPE. The weights for all the triangles are then aggregated. Afterwards the Highest Weighted Area (HWA) is most likely to contain the node. The number of triangles that need to be tested is determined by $\binom{n}{3}$ where n is the number of anchor points. Therefore, we know that ten triangles need to be tested for the example given in Figure 4.3(b). The aggregation of weights can be better understood in a visual manner, illustrated in Figure 4.4.



(a) (b) (c)

Figure 4.4: Illustration of weights being added for APIT tests, with ΔABE having having being tested first (left), then ΔCDE (centre), repeating until all triangles are tested (right).

The weights in this figure are represented as a shade of red, and the IPE as the green dot near anchor C. If we test ΔABE, we see that it contains the IPE. Therefore we add a weight to it, likewise with ΔCDE. Note that

Figure 4.4(b) would not change if we also tested ΔACD because this triangle does not contain the green dot and therefore does not add a weight. We can see from the aggregated result in Figure 4.4(c) that the HWA contains the initial position estimate but, we still need to find the node's position.

We add an additional step to our modified APIT implementation in which we perform SGD to minimise the NLMS error. However, we constrain the position estimates to within the HWA. For the initial position of stochastic gradient descent we use the centroid of the HWA. The centroid being the mean x and y coordinates of all the points within the HWA.

To evaluate the performance of this algorithm we use level 3 filtering on our dataset, leaving us with 6790 transmissions. In Table 4.1 we can see the results which we also compare to the results of KPN's geolocation service for the same data points.

	IPE	NLMS	Mod. APIT	KPN
median position error (km)	1.24	3.16	1.62	0.17
80th percentile error (km)	2.76	17.88	3.72	1.56

Table 4.1: Comparison of the median and 80th percentile position errors for IPE, NLMS, Modified APIT and, KPN's geolocation service.

As we can see from the table, the median position error was 1.62km and the 80th percentile was 3.72km. This is a great improvement from the previous section, showing that the area constraints had the desired affect. However, this method suffers two weaknesses that are inherent to APIT [17]. First, when the node lies near a triangle border, the IPE could be in the wrong triangle. Second, when a node lies outside the area demarcated by the outermost gateways, its location will not be found because triangles cannot be formed outside this area. In both cases the HWA will not contain the node, resulting in large error.

We found that the node's position actually falls outside the gateways' perimeter 36% of the time, a significant amount. This is because of nearby buildings blocking the signals' paths in certain directions. Looking at its weaknesses, it seems the problem inherent in APIT is that the boundaries of the triangles are defined by the gateway positions with nothing to cater for when a node is near the boundary. Therefore, we need an algorithm that uses TDoA measurements to define the border of the highest weighted area so that the node can be located outside the gateway perimeter and, so that the node is less likely to lie on a boundary line, to cater for measurement error. This leads to the development of our localisation algorithm, Multilateral Dissection.

4.5 Multilateral Dissection

This algorithm is based on TDoA, but is a much simpler method that uses an area-based approach. We call our method *Multilateral Dissection*, abbreviated to the acronym MLD.

4.5.1 TDoA Measurement Uncertainty

When considering a hyperbola that represents points with a constant TDoA, we notice that a change in the TDoA measurement affects the turning point of the hyperbola linearly, whereas the error has a pseudo-quadratic effect further along the curve from the turning point.

Let's consider (2.3). After rearranging the equation and setting $y = 0$ we can see the linear relationship between x and the TDoA distance.

$$x = \Delta d/2 \tag{4.11}$$

We can also illustrate the effect of uncertainty visually, if we draw a hyperbola for a certain TDoA (blue) and show the effect of $\pm 10\%$ error in the TDoA measurement (red). See Figure 4.5.

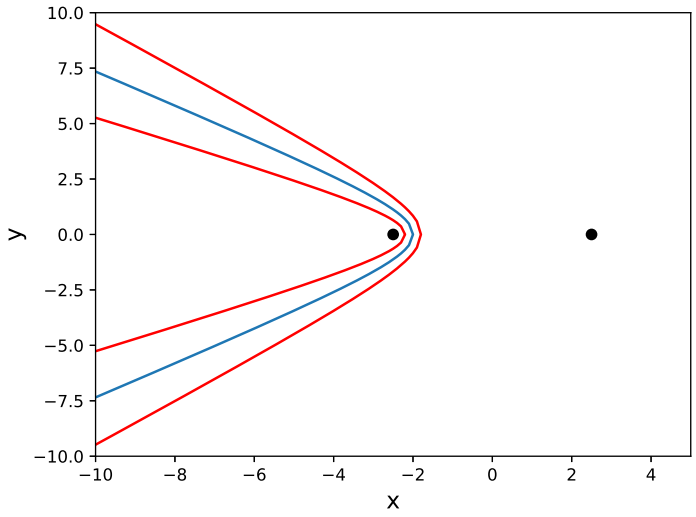


Figure 4.5: Plot showing the effect of 10% TDoA measurement uncertainty.

From this we clearly see that measurement uncertainty in the timestamps has a more drastic effect further away from the turning point. One aspect that we can be fairly certain of is the direction of the curve. We know which gateway the hyperbola will curve towards because we know which gateway received the transmission first. When there is a TDoA measurement of

approximately zero, we cannot be sure which way the hyperbola will curve. Because these hyperbolae are so greatly affected by uncertainty we exclude their TDoA measurements, in addition to level 3 filtering.

4.5.2 Bisection of Environment

If we know which way the hyperbola will curve, we can bisect our environment into where the curve will, and will not be. Furthermore, from (4.11) we know that the hyperbola will have a turning point at $\Delta d/2$ where the environment will be bisected. [1] We will explain how each TDoA measurement dissects the environment using vector calculus.

To start the bisection, we need to consider a pair of anchor points in a two-dimensional plane. Let the position of an anchor, in global coordinates, be represented by $\vec{P} = (X, Y)$ where $\rightharpoonup$ over a variable indicates that it is a vector.

The two anchor points are then at positions $\vec{P}_i$ and $\vec{P}_j$ where $1 \leq i < j \leq n$. We then define $\vec{D} = \vec{P}_j - \vec{P}_i$ where $\vec{D}$ represents the displacement from anchor i to anchor j.

To find the point where the bisection will be, i.e. the turning point of the hyperbola, we move $\Delta d/2$ from the midpoint between the anchor pair, towards the anchor with the earliest timestamp. This is shown mathematically by the following vector addition.

$$\vec{P}_b = \vec{P}_i + \frac{1 - \Delta d}{2D}\vec{D} \tag{4.12}$$

where D is the scalar magnitude of $\vec{D}$ and $\vec{P}_b$ is the point where the line of bisection would intersect the line between the two anchor points. Note that the sign of Δd must be kept from the calculation of $\Delta d = c\Delta t$ as it indicates which anchor point is closer.

A line is then drawn through $\vec{P}_b$, perpendicular to $\vec{D}$, dividing the environment into two parts. The result can be seen in Figure 4.6.

In this figure the green line represents $\vec{D}$. We can also see the red line dividing the environment into where we expect to find the hyperbola and where we do not. The side with the closest anchor point then gets a weight added to it, represented by the shade of red. If an equal weight is added for each anchor pair the shaded areas start to overlap, forming the highest weighted area where the node is most likely to be. An example of this has been shown in Figure 4.7.

The black dots in this figure are the anchor positions, and we can see that the node (cyan coloured dot) falls within the area with the highest weight (darkest shade of red).

[1] Due to measurement error we cannot be completely sure exactly where around $\Delta d/2$ the hyperbola will fall. We will address this issue later in this section.

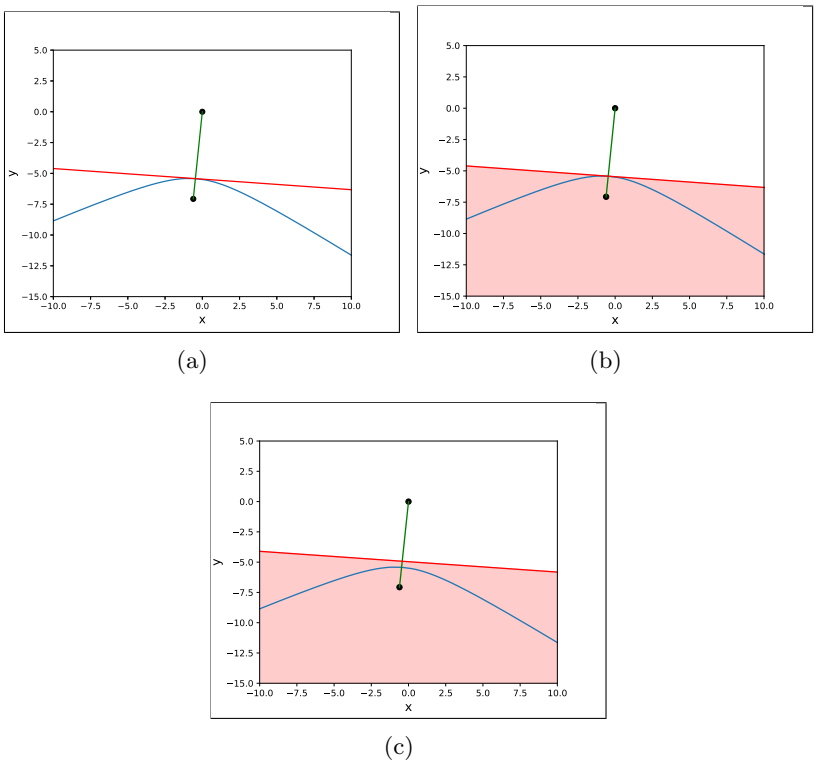


Figure 4.6: Plot showing how the environment is bisected, illustrated by the red line. Shown in (b) is the weight added to the side containing the hyperbola, represented by the shaded area and, (c) illustrates the gap left between the hyperbola and bisection line, to cater for measurement error.

However, due to TDoA measurement uncertainty, a node may not always be within the highest weighted area (HWA). If the node's position falls close to the turning point of a hyperbola, it can sometimes find itself on the incorrect side of a bisection, as the error in it's TDoA measurement causes the hyperbola to be shifted, shown in Figure 4.5. For this reason we have to leave some space between the hyperbola and the bisection line as we show in Figure 4.6(c). The size of the gap was determined empirically, comparing the difference in distances between the node's position and the centroid of the HWA. The performance of MLD with different gap sizes is presented in Table 4.2.

From Table 4.2 we can see that a gap size of 500m had the lowest position error for MLD. Therefore this gap size will be used during implementation. An interesting result from Table 4.2 is that the mean position error of MLD is better than that of KPN. A reason that MLD's mean position error is kept relatively low is because it's an area-based technique. Therefore, as we know from Section 4.4, the position estimations, as well as their error, are always

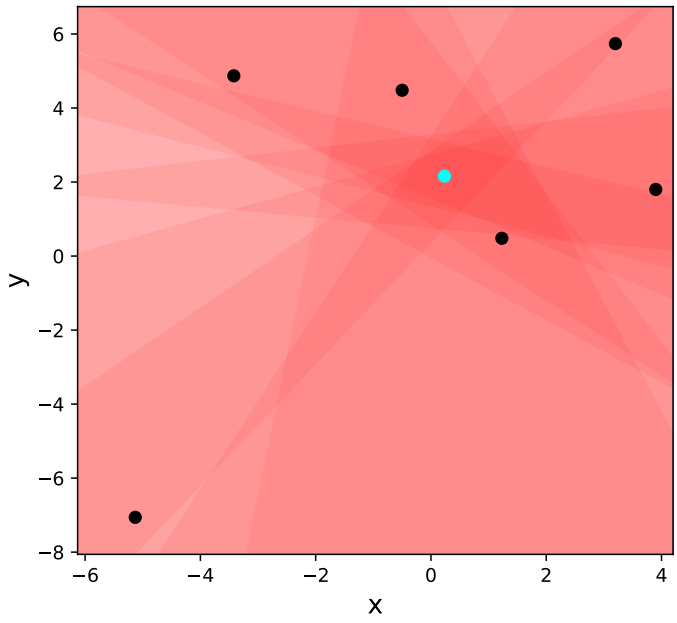


Figure 4.7: Plot showing an example of the aggregated weights for all the receiver pairs of a transmission.

gap size (m)	100	500	1000	1500	2000	KPN
mean position error (m)	2416	2348	2358	2406	2464	6008
median position error (m)	700	661	721	825	948	174

Table 4.2: Comparison of the mean and median position errors for different gap sizes using MLD and, for KPN's geolocation service.

contained, whereas other techniques are more vulnerable to extreme measurement error. Another reason that MLD is not as susceptible to extreme measurement error comes from its use of weighting and bisection. When there is an extreme measurement error, the line of bisection will most likely not be very near the node's position. Let's contemplate this graphically. When a bisection line is far away from the node and its weight is added, it will shade in an area from the bisection line covering a very large area around the node or, it will shade an area far away from the node, depending on the direction of the TDoA hyperbola. By shading a very large area, the added shade of red will contribute to such a large region that it effectively cancels itself out. When a faraway region is shaded in the opposite direction, it doesn't contribute to shaping the HWA. In this manner, many of the extreme measurement errors are ignored.

From Table 4.2 we can also see that the median error was 661m for MLD with a gap size of 500m which, although much lower than our initial estimate algorithm, still has room for improvement when compared to KPN's median error. The 80th percentile is 1.74km for MLD with a gap size of 500m whereas KPN's is only 1.56km.

4.5.3 MLD with SGD

To improve our results, we evaluated the performance of using Stochastic Gradient Descent with Non-linear Least Mean Squares constrained by the HWA from Multilateral Dissection. The initial position for SGD was set as the centroid of the HWA. We used level 3 filtering on our dataset to evaluate this method. The parameters used for SGD were the same as in Section 4.3.

The results of using SGD with MLD were actually not an improvement. The median position error was 715m and the 80th percentile was 1.81km. As we mentioned, MLD tends to be slightly oblivious to large measurement errors. However, when using SGD these measurement errors come back into effect as they will contribute to the LMS error, degrading the algorithms performance. With SGD having failed to improve the results of MLD we conclude that not much more can be improved upon for the core of MLD. Thus we have finished developing the MLD localisation algorithm and, have found its optimal parameters for this dataset. The optimal settings were to use a 500m gap size and simply use the centroid of the HWA to estimate position. With MLD's development completed, we must find a new approach to improve results.

4.6 MLD with Memory

Another well-known localisation technique is dead-reckoning. In this technique, the initial node position is known but, as soon as the node starts moving the position is estimated only using information about the node's movement. In a similar manner, if we have the last known location of a node, we can reduce the likely area that should contain the node, to be a circle around its previous position. The radius of this circle can be determined by the maximum distance that we expect a node to travel, regardless of time. For example, if we consider a person living in Delft, and quite some time has passed without a transmission being sent for whatever reason. It is a fair assumption that the person is still in Delft, i.e. a certain radius from their last position.

To get the maximum expected distance we examined our data with level 1 filtering. We analysed the displacement of each of our nodes between consecutive transmissions. From this analysis we found that in 96% of the cases, a node had moved less than 1km. In the 4% that move more than 1km, it is often the case that many packets have been dropped, which is

the case whenever the battery of a mobile node dies. The small amount of movement between packets could also be due to some of the limitations discussed in Section 3.5. Whatever the case may be, we will use 1km as the maximum radius that a node could have moved from its previous location.

To combine this with MLD, we perform the MLD algorithm as normal to find the HWA and we then plot a circle around the previous position estimate. Where the two overlap is the new, smaller HWA. In Figure 4.8 this can be seen more clearly.

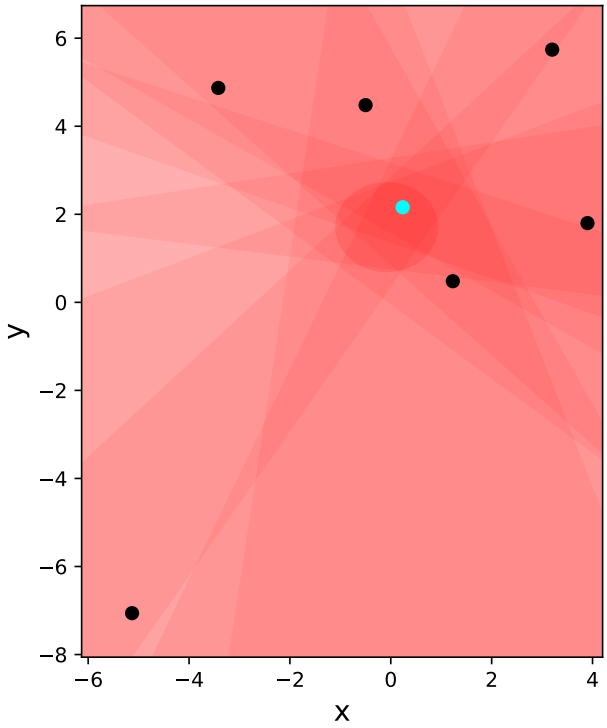


Figure 4.8: A weighted circle (shaded red) is added to Figure 4.7.

Here we can see the circle around the previous position estimate cutting through the HWA. When implementing this algorithm, there could be a case where the HWA does not intersect the circle. In this case, we could either use the HWA or the previous position estimate. Due to the fact that most nodes move very little between transmissions, we have chosen to use the previous position estimate in cases where there is no overlap. However, if there is no overlap for the next localisation as well, we then assume that the node has in fact moved more than 1km from the previous position. In this case we will use standard MLD without memory for this estimation. We

do this to ensure that an estimate does not get left behind if the node has moved. For the subsequent transmissions we continue using the algorithm with memory.

We evaluated this algorithm using our dataset with level 3 filtering. However, in this case we had to sort the data, making sure that a node's transmissions were grouped together and, were in chronological order. The median position error of this algorithm was 500m with 80% of cases having an accuracy of less than 925m. We feel that this is a great improvement because if we compare this to KPN's geolocation service for the same transmissions, although their median error is lower, KPN only achieves an 80th percentile of 1558m. We will show the CDF's for our results in Section 4.8 at the end of this chapter.

Inspired by these results, we decided to attempt one last adaptation of our MLD algorithm to improve our results further.

4.7 Asymptotic Division

Seeing as the idea behind MLD is to find a general area where we expect to find the node, we thought that we can refine this area while still allowing for measurement uncertainty. To do this, we use the asymptotes of the TDoA hyperbolae so that each weight added by a TDoA measurement is more conical as opposed to a flat area as it was in Section 4.5. The asymptotic dissection is illustrated in Figure 4.9.

There are two things that we should notice when looking at this figure. Firstly, the asymptotes don't come to a point but are actually cut off by the same bisection line that was used in standard MLD. This was done so that the weighted area remains as small as possible. The gap-size for this line is still 500m. The second point to notice is that the asymptotes, represented by the dashed line, slightly diverge from the hyperbola instead of the hyperbola tending towards the asymptote. This is because the asymptote relates to a TDoA measurement that is 10% less than the actual measurement. This is to cater for measurement uncertainty. Note that the asymptotes do not have an added gap as the bisection line does. The 10% uncertainty is simply added into (2.4) when plotting the asymptotes of the hyperbola. The uncertainty value of 10% was determined empirically.

When we were determining how much uncertainty to compensate for, we found an interesting change in performance relating to a change in uncertainty. The results performance of different asymptote uncertainties is shown in Table 4.3. Here 100% uncertainty results in the asymptotes being a flat line again which is the same as implementing MLD normally.

As the uncertainty value was increased more and more towards making the asymptotes a flat line, the median position error also increased. However, the 80th percentile position error actually decreased, although this is less

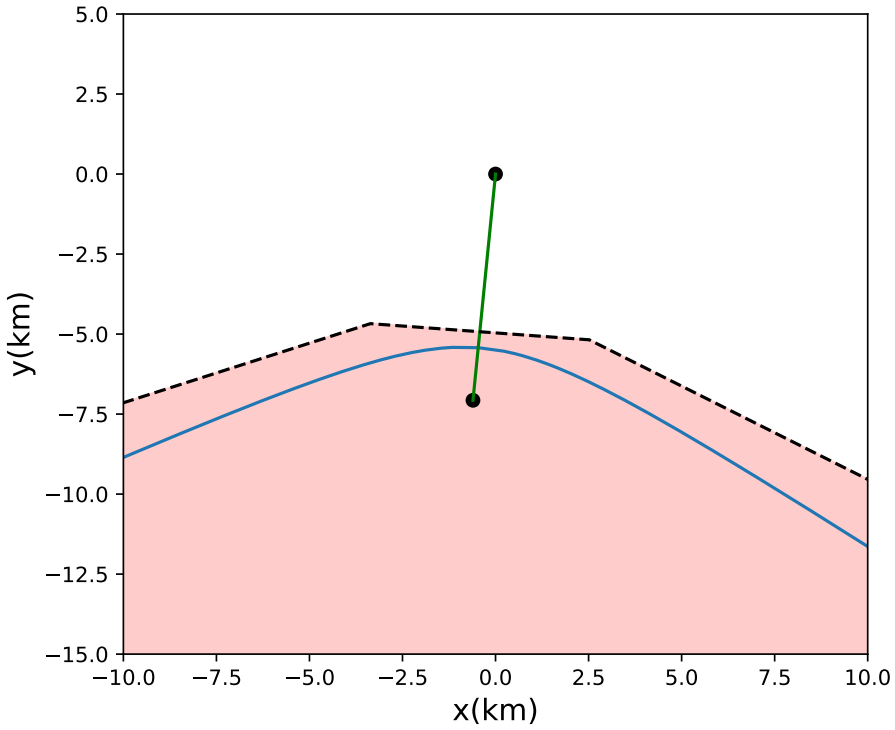


Figure 4.9: Illustration of how the asymptotes of the hyperbola are used to divide the area.

apparent at lower uncertainties. The inverse was also true; as the uncertainty decreased the median error decreased as well until the uncertainty reached roughly 10%, where the median error reached its lowest point. This actually relates somewhat to the accuracy of our TDoA measurements, as a low uncertainty in the asymptotes will mean that the HWA will be much more refined thus, more accurate. This is why the lower percentiles decreased in error as the median error did. However, when there are TDoA measurements with a larger error, these will fall outside of the asymptotes if they only compensate for a small uncertainty. Hence, the increase in error at the higher percentiles. This also explains why the median error increases again at 0% uncertainty, because at this point the uncertainty compensation is so low that even TDoA measurements with low error are likely to fall outside of the asymptotes.

We evaluated this algorithm using the memory technique from the previous section and using level 3 filtering on our data set. The change in performance due to the uncertainty is evident when we consider the dif-

% uncertainty	0	10	20	30	100
median position error (m)	544	481	487	488	500
80th percentile error (m)	1224	1118	1104	1111	925

Table 4.3: Comparison of the median and 80th percentile position errors different asymptotic uncertainty compensations.

ference in results between 100% asymptote uncertainty (standard MLD) and 10% uncertainty. Using 10% as the final uncertainty parameter, this algorithm has a median position error of 481m and an 80th percentile of 1118m. When comparing to 500m median error for standard MLD with memory, we feel that the small change in performance does not compensate for the added complexity involved in calculating the asymptotes and their intersection points with the line of bisection, especially when it performs worse for higher measurement uncertainty.

We do not see this as a failure but instead it gives some insight and serves to highlight the advantages that MLD has with regards to TDoA measurements with larger uncertainty. However, seeing as it is not a significant improvement we will not consider it further. In the next section we will give a summary of this chapter with a comparison of the different algorithms.

4.8 Comparison of Algorithms

In this section we compare the results of the different localisation implementations. We will compare the performance of our initial position estimation algorithm, the algorithm from [16] using their NLMS approach, the modified version of APIT and of course, our MLD algorithm, with and without memory. We will also compare these to the performance of KPN. For all implementations we used the same LoRa transmissions. Once again, the transmissions were selected using level 3 filtering which left 6790 transmissions. The results are plotted in a CDF shown in Figure 4.10.

In Figure 4.10 we can see how the algorithms progressed over the course of development. Some localisation algorithms are not shown here. These include, trilateration with RSSI which performed too poorly to be comparable, MLD using SGD which was excluded as SGD in fact decreased the accuracy of MLD, and lastly, MLD with asymptotic division is not shown because it did not show significant improvement.

Quite surprising from our results, was the performance of our initial position estimation algorithm, which initially was only intended to serve as a starting point for other algorithms but, ended up performing better than these algorithms. Looking at the area-based techniques, we see that they all seem to perform rather well in the higher percentiles of the CDF, including the modified version of APIT which manages to surpass KPN's performance

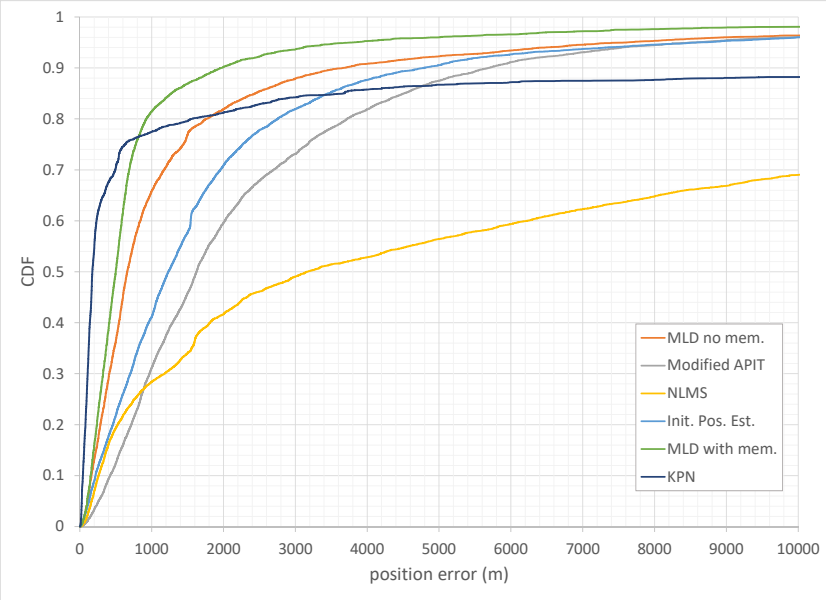


Figure 4.10: Comparison of the position error of the localisation algorithms that were implemented.

in the 86th percentile. Our initial position algorithm also performs rather well in the higher percentiles, the reason for which is unknown. The best of the area-based techniques is MLD localisation with memory, which had a median position error of 500m and approximately 82% of cases with under 1km position error, showing the effectiveness of area-based techniques for mitigating large measurement error.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this work we have implemented multiple localisation algorithms on a proprietary LoRa network using mobile LoRa devices. As far as we know, no work has been published that evaluates localisation algorithms on an LPWAN over such a large urban area. Research into Localisation in a LPWAN so far, has focused on trilateration with RSSI. We have evaluated the feasibility of using RSSI ranging with LoRa and determined that it is unreliable over a large urban area, due to the increased interference.

We developed a weighted-centroid localisation algorithm that uses timestamps for weighting. However, our main contribution is the novel Localisation technique that we invented, called Multilateral Dissection (MLD), an area-based technique that uses TDoA measurements. Using this method and a modified version of APIT, we showed that area-based localisation algorithms can be used to reduce the effect of large measurement error by restraining the region where position estimations can be.

We also evaluated the performance of MLD with various adaptations. Besides the standard MLD, we tested a variation that used the asymptotes of a TDoA hyperbola which gave some interesting insights into TDoA uncertainty although these will need to be investigated further. Another MLD variation used Non-linear Least Mean Squares with Stochastic Gradient Descent. MLD with SGD performed worse than standard MLD indicating that MLD mitigates the effect of large measurement error because of its weighting strategy.

Adding to this, we improved MLD's accuracy with a variation that uses the previous location estimation to refine the area where the node is likely to be. We showed that this algorithm has a higher localisation accuracy on a LoRa network than using Stochastic Gradient Descent with Non-linear

Least Mean squares, which is one of the algorithms that is traditionally used for TDoA localisation with other radio technologies. We showed that although MLD has a higher median error than KPN's geolocation, by using previous position estimates, MLD has a better mean performance, with 80% of cases being located to within 925m of the node's location, compared to 80% of KPN's cases being located to within 1558m.

5.2 Future Work

LoRa wireless communication technology is a proprietary technology that is also relatively new. These two factors mean that there is still much research that can be done, especially for localisation as there is currently a lack of public research in this field. Although this work presents some localisation solutions for LoRa networks there is still much research that can be done to improve them. We propose the following topics that we feel will improve localisation in LoRa networks:

- **Multipath Detection** — One of the biggest sources of error for localisation in urban environments is multipath effects. If research is done to detect if a signal was affected by multipath and more importantly, how to mitigate the error from multipath, then the accuracy of localisation will improve immensely.

- **Environment Mapping** — By mapping the environment, there is so much that information that could be used. This could tie in with the point above, to help estimate multipath, but it could also be used to estimate the range of a signal, or at the very least it could be used to find the most likely location that you would be based on your surroundings. For example, if you were walking you most likely wouldn't be located in a river.

- **Machine Learning** — Although we were not able to properly implement any machine learning techniques in this work, there is already some research into using machine learning for localisation in LPWANs. However, there is not a great deal as of yet. So far the use of Support Vector Machines has been shown to improve results and we feel that further research into this field can improve localisation even further.

- **Finger-Printing** — Finger-printing is a localisation technique whereby signal strength measurements are taken from ideally all locations where a device needs to be localised. This is quite difficult to do on a scale as large as LoRa is capable of but, using crowd sourcing for data collection, this could be possible which is being already being realised

by The Things Network with their TTN Mapper database. Fingerprinting may not be extremely accurate on its own but, when combined with other techniques, it will most likely improve localisation results.

- **Motion Models** — The memory feature that we added to MLD improved the accuracy by a significant amount. However, this just focused on a 1km radius around the previous position. By building an accurate motion model, the direction and distance could be more accurately predicted which should in turn improve the accuracy of localisation. Research into this topic would tie in well with Environment Mapping which we mentioned earlier in this section.

Bibliography

[1] Lora — geolocation. https://zakelijkforum.kpn.com/lora-forum-16/lora-geolocation-8555, 2016. [Online; accessed 8-August-2018].

[2] Map and tile coordinates. https://developers.google.com/maps/documentation/javascript/coordinates, 2018. [Online; accessed 12-September-2018].

[3] Shadowview and internet of life continue as smart parks. https://www.smartparks.org/news/shadowview-and-internet-of-life-continue-as-smart-parks/, 2018. [Online; accessed 16-September-2018].

[4] Sigfox geolocation, the simplest and cheapest iot location service. https://www.sigfox.com/en/sigfox-geolocation, 2018. [Online; accessed 30-August-2018].

[5] Great Britain. Admiralty. *Manual of Navigation*, volume 1. HM Stationery Office, 1967.

[6] Christophe Adrados, Irène Girard, Jean-Paul Gendner, and Georges Janeau. Global positioning system (gps) location accuracy improvement due to selective availability removal. *Comptes Rendus Biologies*, 325(2):165–170, 2002.

[7] Qasim Zeeshan Ahmed, Wei Liu, and Lie-Liang Yang. Least mean square aided adaptive detection in hybrid direct-sequence time-hopping ultrawide bandwidth systems. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 1062–1066. IEEE, 2008.

[8] M Bavaro, D Galliano, A Annunziato, and J Fortuny-Guasch. Gps sea level measurement device; doi:10.2788/785155. Technical report, European Commission, 2016.

[9] Jan Blumenthal, Ralf Grossmann, Frank Golatowski, and Dirk Timmermann. Weighted centroid localization in zigbee-based sensor networks. In *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pages 1–6. IEEE, 2007.

[10] Martin C Bor, Utz Roedig, Thiemo Voigt, and Juan M Alonso. Do lora low-power wide-area networks scale? In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 59–67. ACM, 2016.

[11] Nirupama Bulusu, John Heidemann, and Deborah Estrin. Gps-less low-cost outdoor localization for very small devices. *IEEE personal communications*, 7(5):28–34, 2000.

[12] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67, 2016.

[13] Sinan Gezici, Zhi Tian, Georgios B Giannakis, Hisashi Kobayashi, Andreas F Molisch, H Vincent Poor, and Zafer Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE signal processing magazine*, 22(4):70–84, 2005.

[14] Samir Goel, Tomasz Imielinski, and Kaan Ozbay. Ascertaining viability of wifi based vehicle-to-vehicle network for traffic information dissemination. In *The 7th International IEEE Conference on Intelligent Transportation Systems*, volume 3, pages 1086–1091. Washington DC: IEEE Press, 2004.

[15] Juergen Graefenstein and M Essayed Bouzouraa. Robust method for outdoor localization of a mobile robot using received signal strength in low power wireless networks. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 33–38. IEEE, 2008.

[16] Fredrik Gustafsson and Fredrik Gunnarsson. Positioning using time-difference of arrival measurements. In *ICASSP (6)*, pages 553–556. Citeseer, 2003.

[17] Tian He, Chengdu Huang, Brian M Blum, John A Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95. ACM, 2003.

[18] Pascal Jörke, Stefan Böcker, Florian Liedmann, and Christian Wietfeld. Urban channel models for smart city iot-networks based on empirical measurements of lora-links at 433 and 868 mhz. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*, pages 1–6. IEEE, 2017.

[19] Matthew Knight and Balint Seeber. Decoding lora: Realizing a modern lpwan with sdr. In *Proceedings of the GNU Radio Conference*, volume 1, 2016.

[20] Kolff Computer Supplies BV, Dordrecht, Netherlands. *Datasheet TM-901 PCB-N1C2-V2017-10*, October 2017.

[21] Ka-Ho Lam, Chi-Chung Cheung, and Wah-Ching Lee. Lora-based localization systems for noisy outdoor environment. In *Wireless and Mobile Computing, Networking and Communications (WiMob),*, pages 278–284. IEEE, 2017.

[22] In Lee and Kyoochun Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431–440, 2015.

[23] LoRa Alliance, San Ramon, CA. *LoRaWAN What is it? A technical overview of LoRa and LoRaWAN*, November 2015.

[24] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. A comparative study of lpwan technologies for large-scale iot deployment. *ICT Express*, 2018.

[25] Michael Meuser. Netherlands arcgis shapefile map layers. `http://www.mapcruzin.com/free-netherlands-arcgis-maps-shapefiles.htm`, 2018. [Online; accessed 11-August-2018].

[26] NASA Goddard Space Flight Center, Ocean Biology Processing Group. *Ocean Science Data Product Format Specification*, November 2017.

[27] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 2926–2931. IEEE, 2001.

[28] Lennart Nordin. Spreading factor (sf), time on air and (adaptive) data rate. `https://zakelijkforum.kpn.com/lora-forum-16/spreading-factor-sf-time-on-air-and-adaptive-data-rate-10908`, 2018. [Online; accessed 09-September-2018].

[29] Juha Petajajarvi, Konstantin Mikhaylov, Antti Roivainen, Tuomo Hanninen, and Marko Pettissalo. On the coverage of lpwans: range evaluation and channel attenuation model for lora technology. In *ITS Telecommunications (ITST), 2015 14th International Conference on*, pages 55–59. IEEE, 2015.

[30] Andri Rahmadhani. Performance evaluation of lorawan: From small-scale to large-scale networks. MSc thesis, Technical University of Delft, November 2017.

[31] Andri Rahmadhani and Fernando Kuipers. When lorawan frames collide. 2018.

[32] Gil Reiter. Wireless connectivity for the internet of things. *Europe*, 433:868MHz, 2014.

[33] Hazem Sallouha, Alessandro Chiumento, and Sofie Pollin. Localization in long-range ultra narrow band iot networks using rssi. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[34] Semtech, Camarillo, CA. *Low Energy Consumption Design*, July 2013.

[35] Semtech, Camarillo, CA. *SX1272/73 Datasheet*, March 2017.

[36] Semtech Corporation, Camarillo, CA. *AN1200.22 LoRa Modulation Basics*, May 2015.

[37] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. A survey on lpwa technology: Lora and nb-iot. *Ict Express*, 3(1):14–21, 2017.

[38] John Parr Snyder. Map projections used by the us geological survey. Technical report, US Government Printing Office, 1982.

[39] Thomas Telkamp and Slats Laurens. Ground breaking world record! lorawan packet received at 702 km (436 miles) distance. `https://www.thethingsnetwork.org/article/ground-breaking-world-record-lorawan-packet-received-at-702-km-436-miles-distance`, 2018. [Online; accessed 30-August-2018].

[40] Tim Van Dam. Protecting wildlife with lorawan. The Things Conference, Amsterdam, 2018.

[41] Eric W Weisstein. Great circle. 2002.

[42] Guochang Xu and Yan Xu. *GPS: theory, algorithms and applications*. Springer, 2016.

Appendix A

Derivation of TDoA Hyperbola

To calculate the formula for the hyperbola representing all possible node positions for a given TDoA measurement, we followed the derivation given by Gustafsson and Gunnarsson in [16]. However, to keep the explanation in Section 2.3.3 brief we only gave the final formulae for the hyperbolae and asymptotes. Here, we will show how these formulae were derived by Gustafsson and Gunnarsson in [16].

We define $\Delta d_{i,j}$ as the distance relating to the TDoA measurement via the speed of light:

$$\Delta d_{i,j} = c(\tau_j - \tau_i), \qquad 1 \leq i < j \leq n \tag{A.1}$$

where τ_i and τ_j are the timestamps at anchor point i and j respectively and n is the number of anchor points. Each $\Delta d_{i,j}$ will relate to positions (x,y) on a hyperbola.

Let's use the same assumptions that we made in Section 2.3.3, where the two anchor points corresponding to a TDoA measurement, lie on the x-axis at $x = D/2$ and $x = -D/2$, where D is the actual distance between the two anchor points.

We can calculate the distances between the gateways and a node's position (x,y) using Pythagoras' theorem. The distances between the node and gateways one and two respectively are calculated as follows [16]:

$$d_2 = \sqrt{y^2 + (x + D/2)^2} \tag{A.2}$$
$$d_1 = \sqrt{y^2 + (x - D/2)^2} \tag{A.3}$$
$$\tag{A.4}$$

Therefore, we can form the equation for Δd as:

$$\Delta d = d_2 - d_1 = h(x, y, D) \tag{A.5}$$
$$= \sqrt{y^2 + (x + D/2)^2} - \sqrt{y^2 + (x - D/2)^2} \tag{A.6}$$

where $h(x, y, D)$ is the formula for all possible node positions (x,y) for a constant, ideal TDoA distance, where the anchor points are a distance D apart from each other.

We can then rewrite the equation in the form of a hyperbola [16] which we presented in Section 2.3.3:

$$\frac{x^2}{a} - \frac{y^2}{b} = \frac{x^2}{\Delta d^2/4} - \frac{y^2}{D^2/4 - \Delta d^2/4} = 1 \tag{A.7}$$

To find the asymptotes of this hyperbola, we can solve this equation in terms of y for the case when $x \gg \Delta d^2/4$ and $y \gg D^2/4 - \Delta d^2/4$ [16]:

$$y = \pm \frac{b}{a} x = \pm \sqrt{\frac{D^2/4 - \Delta d^2/4}{\Delta d^2/4}} x \tag{A.8}$$