ASHESI UNIVERSITY COLLEGE

DETERMINING THE LOCATION OF A BUS IN REAL-TIME USING LORA TECHNOLOGY

CAPSTONE
B.Sc. Electrical and Electronics Project

RONALD TUMUHAIRWE
2019

ASHESI UNIVERSITY COLLEGE

DETERMINING THE LOCATION OF A BUS IN REAL-TIME USING LORA TECHNOLOGY

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Electrical and Electronics Engineering.

RONALD TUMUHAIRWE

2019

DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……………………………………………………………………………………………

Candidate's Name: Ronald Tumuhairwe

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University College.

Supervisor's Signature:

……………………………………………………………………………………………

Supervisor's Name: Nathan Amanquah

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

Acknowledgements

To my supervisor, Dr Amanquah Nathan whose encouragement and academic advice helped me undertake this project. Mr.Nicholas Tali for helping me to select the appropriate equipment to use in this project. To all the people whose encouragement and academic advice helped me undertake this project. I would like to sincerely thank you for helping me to make this project a success.

Abstract

Travelers without prior information about the bus they are waiting, often waste time at bus stations which affects their day activity plan. GPS receivers are used to address this problem by providing real-time about the location of the bus, however, it is quite expensive and power consuming to use and every bus needs to have a GSM/GPRS module to be able to send the information to the internet. This project suggests a low-cost and less power consuming approach of using LoRa technology, geometry methods to determine the location of the bus and only one GSM module for multiple buses to transmit the data to the internet. Results in the project show success in determining the location of the bus in real-time.

Table of Contents

Table of Tables

Table of Figures

1. Introduction

Most people go to stations to wait for buses to their destinations. However, it is quite frustrating when the bus takes a time longer than expected to arrive. There is thus a need for a suitable technology to provide passengers with real-time information about the current location of the buses. The commonly used technology for tracking buses in real-time is Global Positioning System (GPS) coupled with communication technologies like cellular internet access for example 2G,3G. However, this project suggests a cheaper and more energy saving method using a wireless long-range communication technology 'LoRa' and geometry to determine the current location of the bus to enable tracking of the buses.

1.1 Background

Travelers usually get frustrated while waiting for buses at stations without any certainty about how long the bus will take to arrive at the station. This leads to time wastage and anxiety especially when the traveler did not consider in his or her plan that the bus may delay than usual. Empirical evidence suggests that the time travelers spend at stations waiting for transportation is more tiresome than the time they spend inside the bus in motion to their destination. This is largely due to the higher degree of uncertainty associated with waiting for a transit bus. Knowing the current location of the bus will help travelers to manage their time appropriately before boarding buses which will improve passenger satisfaction and ultimately increase bus ridership [1].

1.2 Problem statement

Travelers without prior information about the bus they are waiting, often waste time at bus stations which affects their day activity plan.

1.3 objectives

The objectives of this project;

- Designing a bus tracking system that will enable travelers know the current location of the bus in real-time.
- Using LoRa technology and geometry methods to determine the location of the bus.
- Designing a web application that will display the different buses available in relation to the station they are located.
- Storing information in a database server about the different places the buses have moved to during the day.

1.4. Justification/ Motivation of Project Topic

A survey aimed at examining users' views towards the availability of real-time bus information survey was conducted in Calgary, Alberta, Canada. Results showed that 35.5% of the respondents agreed that it is frustrating to wait for buses at stations without prior information on how long it will take the bus to reach the station [2]. Another survey to observe passengers' waiting behaviors without prior information about the bus arrival time at six bus stops operated by the Bureau of Transportation, Tokyo Metropolitan Government was conducted. It revealed that over 50% of the travelers sat on benches doing nothing which showed a complete waste of time. The irritation levels are even higher when the facilities at the waiting places are not very favorable. for example; no benches, no shades and sometimes dust coming from the roads [3]. Lack of real-time information about the bus proves to be a big problem for travelers and thus this gap has to be bridged to improve on the user experience and also enable travelers to spend their time more productive other than sitting at the stations without prior information about the bus they are waiting for. This project seeks to address this challenge by providing real time information about the current location of the bus by giving travelers a chance to adjust their plans accordingly. The solution includes using LoRa technology between the bus and three base stations. Geometry methods and algorithms are used to determine the location of the bus based on the time of arrival of the regular signals sent from the bus to the various base stations.

1.5. Project Requirements

The bus should be able to send frequent signals to the base stations and the base stations should be able to pass on the time of arrival of these signals to the Web server where computations to determine the current location of the bus are performed. Ultimately, travelers should be able to determine the current position of the buses they are waiting for at any particular moment. The LoRa transceivers and geometry methods should be able to determine the current location of the bus within 15km as the maximum distance from each base station. The traveler should be able to use a web application to determine the different buses on a map within the vicinity of the station. The user should be able to select a desired bus and the estimated time of arrival will be displayed on the screen. The project should provide relatively accurate results and updated information of the current location of the time of arrival of the signal.

2. Literature Review

Several technologies have been introduced to help travelers track buses in the city by providing information on their devices or stations about the current location of the buses and the routes they are taking. The most common innovation 'Google maps', a web-based service that provides detailed information about geographical regions and sites around the world [4], is one way that travelers use to trace the route they should follow to their destination. Google maps now has updates for Android users that incorporate real-time mass transit data, letting users not only to plot their bus routes but also tracking them in real time. These updates are currently available only in six cities worldwide: Boston, Portland, San Diego, San Francisco, Madrid and Torino and has not yet been integrated in the public transit. The app has an option for selecting a preferred station and returns the pinpoint of the location assisting the user to know when the next bus is expected and the amount of time so that he or she plans his or her time accordingly prior to boarding the bus [5]. Google maps uses satellite imagery and street maps provided by Google Earth which takes high resolution photographs of the planet. Google maps uses GPS to make accurate time measurements and to determine the current coordinates of the user in order to give real-time traffic updates to the user based on location [6]. Cellular data is used to provide internet connection on the user's device to access Google map services.

Several projects have been embarked on to ease the stress that travelers go through waiting for buses at stations. To make sure travelers are able to get real time information about public transport like buses at stations, wireless communication has been used intensively. This is mainly because most cities are quite huge, for example a city like Accra which has an area of 238,540 square kilometers [7], if data is be transferred across such wide areas. Wireless communication will be ideal because signals travel over free space, users can move freely within the area of the network with their devices and there is no limitation to number of connection ports [8] for users in the city. A commonly used wireless communication networks is cellular networks.

Wireless communication has proved to be one of the most efficient ways of connecting two or more devices to transfer information. Currently a lot of technologies exist including; Bluetooth, Wi-Fi, GPRS, Satellite, microwave, Infrared Radiation (IR), High Frequency (HF), among others. Each of these technologies is designed to suit certain applications. The factors considered include coverage range, the amount of data that can be transmitted at unit time, cost among others. Over time, new technologies have further emerged for long range applications and the most significant one that has solved the 'high power consumption' problem faced by GPRS is called LoRa. LoRa is a modulation based on a variation of chirp spread spectrum with integrated forward Error correction. This allows LoRa to trade off data rate for sensitivity with a fixed channel bandwidth by selecting the amount of spread used (a selectable radio parameter from 7 to 12) [9]

The evolution of mobile communication technologies has also improved wireless communications. Mobile communication technologies can be categorized into generations which include 1G, 2G, 3G and 4G where G stands for Generation. The first generation used analog transmission and modulated to a frequency of 150 MHz and above. The Second generation used digital transmission for voice signals. Digital technology provides improved battery life for wireless phones and added many features like caller ID, digital encryption, text messaging and intelligent roaming. In this network, data is usually transported over voice channels at speeds ranging from 9.6Kbps to 14Kbps. The third generation made video calling and seamless streaming of video possible and the speed of packet data transport through the network increased to speeds above 100Kbps [10]. Table 1 below shows examples of cellular networks in different generations and some of their characteristics. The data rates are improved in each generation however, this drains the battery of the device faster which is not ideal for battery powered devices.

Table 1: shows examples of cellular networks and their characteristics

Generation	1G	2G	3G	4G
Examples	Global System Mobile communication (GSM)	General Packet Radio Service (GPRS)	Universal Mobile Telecommunications System (UTMS)	Long Term Evolution (LTE)
Data rates	14.4 Kbps	26.8 – 53 Kbps	384 – 128Kbps	300 Mbps
Uplink frequency	880 – 915 MHz	800 – 1900MHz	850 – 2100 MHz	1700 – 2100MHz
Downlink frequency	925 – 960 MHz	800 - 1900MHz	850 – 2100 MHz	2500 – 2700MHz

All these long-range technologies have advantages in terms of data rate speeds and coverage. Cellular networks have fast data rates and high coverage.

2.1 LoRa

In this project, Long Range (LoRa), a long-range wireless platform that can be used in small IoT applications [11] is used.

LoRa technology has the following advantages;

- It uses license-free radio frequency bands.
- It can cover upto 15km which is quite good coverage for tracking buses.
- Consumes low power compared to GSM and other long-range wireless platforms.
- it can be used by public, private or hybrid network [12].

In this project, it is much cheaper to use LoRa modules, GPS, and geometry to determine the location of the bus and then use GSM to transmit this information to a web server compared to using only GSM modules and GPS to determine the location of the bus. It should be noted that LoRa has data rates of 27 Kbps which is significantly slower than newer versions of the cellular networks. This limits the application of LoRa to ‘light weight’ applications in terms of data rates.

2.1 Tracking

Tracking services enable people to know where their belongings, events, among others can be found easily without wasting hours of ‘guess’ searching. In this era, where technology advancement has taken lead, innovations have come up that enable people to keep track of different things easily in real-time. Tracking can be as basic as using short range wireless communications to search for objects in a home. However, tracking applications can also be quite complex and extend to several kilometers to determine the location of objects. In this case, short range wireless communication technologies will not be helpful which makes long range wireless communication technologies ideal for example tracking buses in a city.

Several projects for bus trackers use the Global Positioning System (GPS) to provide coordinates of the current location, a global navigation satellite system that uses atleast 24 satellites, a receiver and algorithms to provide location [13]. These coordinates are transmitted wirelessly to the traveler’s device who is interested in the current location of the bus in real time. The most used wireless networks are cellular networks like GSM and GPRS. Using these technologies with GPS for real time information tracking has the following challenges. The systems require a lot of energy which is not suitable for battery powered devices and data charges are incurred every time, they are accessed [14]. Battery life is very crucial for devices that are currently used which has led to the introduction of more efficient technologies which consume less power yet yielding the same results. Figure 1 below shows a GPS bus tracking system which uses the GSM/GPRS network to share the information with users [15]. A GPS receiver is installed in the school bus which receives current coordinates from atleast three GPS satellites, the coordinates are sent to a server using the GSM/GPRS network. In the server, the coordinates are used as inputs in geometry algorithms to determine the current location of the school bus. Using web sockets, the information is then shared to mobile or web applications for the user to access the location in a more understandable way for example seeing buses as pins in google maps. It should be noted that three or more satellites are needed to determine the location of the bus as explained later.

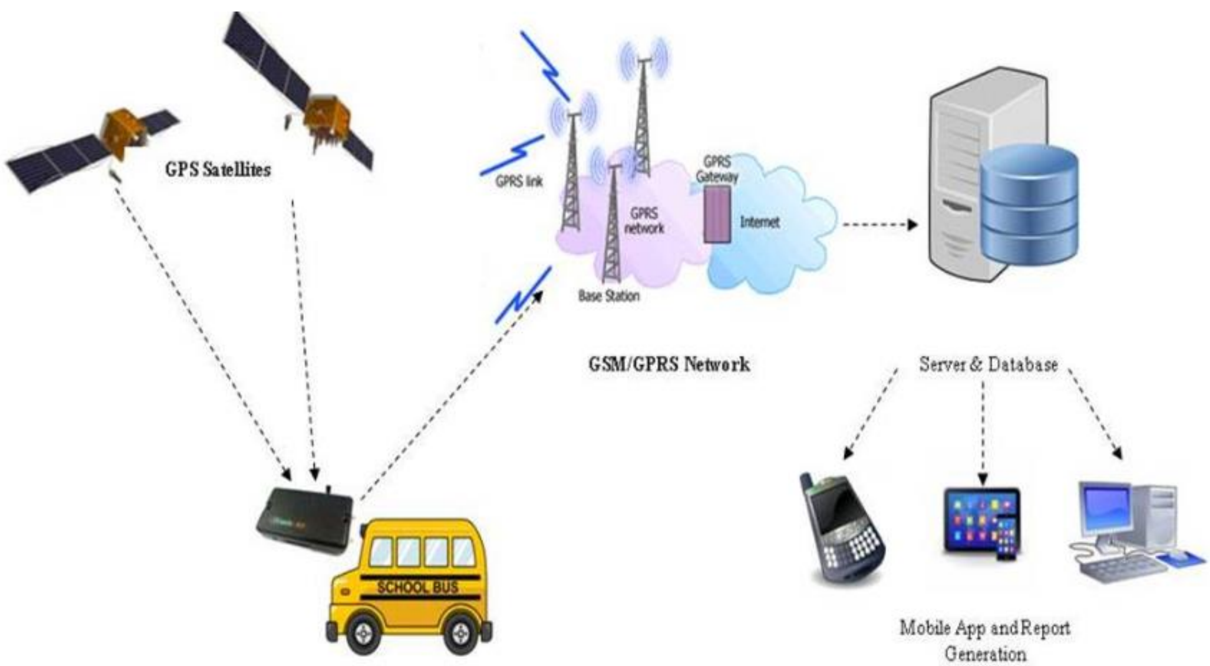

Figure 1:shows a GPS bus tracking system using GSM/GPRS Network to share the information with the user

2.3 Location Determination and algorithms

The bus sends out regular LoRa signals by the LoRa transmitter. The signals are received by base stations located at different points containing LoRa receivers. The distance covered by a

LoRa signal from the bus to the base station can be calculated by multiplying the time taken by the signal to arrive at the base station with the speed of light; this is because LoRa signals are electromagnetic waves which travel at the same speed as light waves. The radius of each imaginary circle is the calculated distance from the base station to the bus. The location of the bus can then be determined from the intersection imaginary circles can be drawn from the base stations to the bus as shown in figure 2 below.

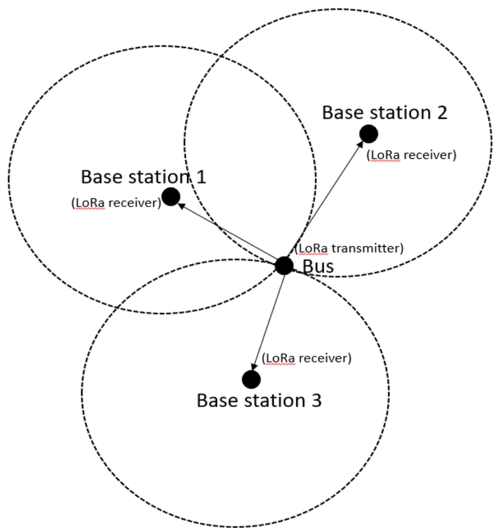

Figure 2: shows how geometry methods can be used to determine the location of the bus as it moves relative to the base stations

2.3.1 Base stations

Base stations consist of different electronic components and antennas and can be located on masts, rooftops, inside or outside buildings. Base stations are deployed for two reasons; coverage and capacity. Base stations should be in a maximum radius of 15 kms from the bus for coverage because they contain the LoRa receivers that are used to determine the location of the bus. The base stations collect time data periodically and send it to the web server. To calculate the location of the bus atleast three base stations should be within the radius of the bus as explained later in the geometry methods. Figure 3 [16] below shows the LoRa gateways that act as base stations to send information from the LoRa transmitters to the cloud server using wireless connection networks.

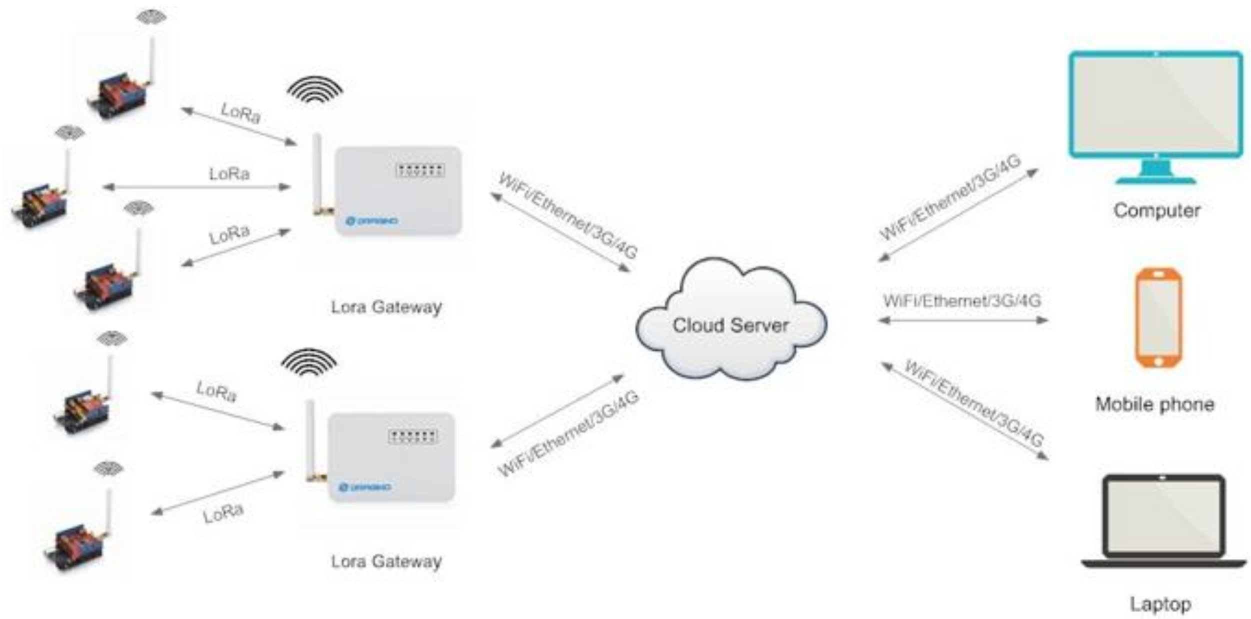

Figure 3: shows LoRa Gateways which act as base stations to send information from the LoRa transmitters to the cloud server

2.3.2 Geometry methods

Geometry methods can be used to determine the location of the bus when the coordinates of the base stations are known as described later in this paper. The location of atleast three base stations needs to be known. If we know only one radius of the bus from the base station, the circle drawn contains a lot of points which indicates that bus can be anywhere on the circumference of circles. Knowing the radius of a second base station from the bus will also not be sufficient because the intersection of the two circles will create two points which means the bus can be at either point. However, knowing the radius of the third base station from the bus will create a third circle. The intersection of these three circles will create one unique point which will be the location of the bus. In real-life situations, the bus may not be at the intersection of one common point due to imperfections of radio transmission and signal distortions caused along the transmission of the LoRa signal from the bus to the base stations such errors include time delay, signal attenuation caused by noise, physical barriers among others. Such scenarios are shown in figure 4 below.

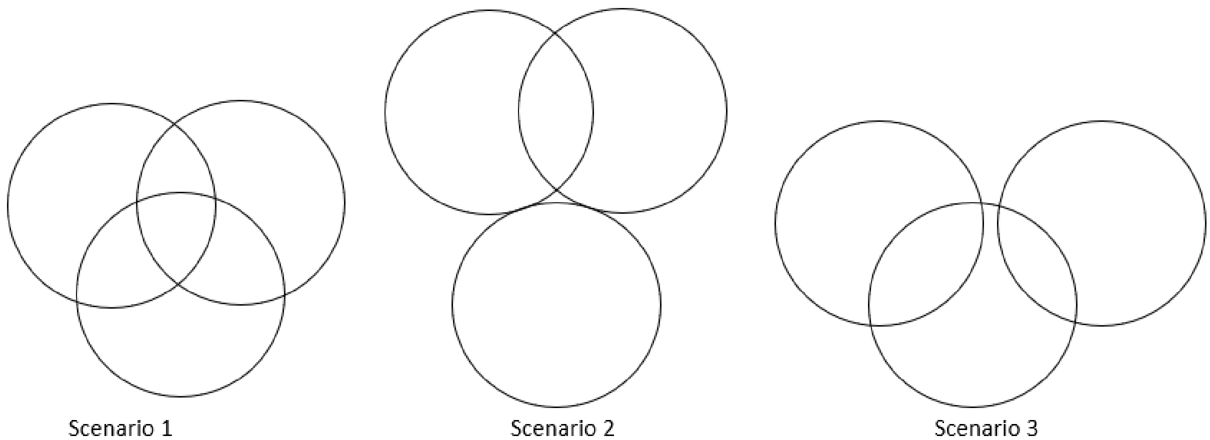

Figure 4: illustration of different scenarios of improper circle intersections due to errors

This project works on the assumption that the circles meet perfectly and only one point of intersection exists among the three circles.

2.3.3 Real Time Location System Implementation Methods

There are a number of different methods of implementing Real Time Location Systems using wireless schemes, but they categorized into two basic types;

- Radio Signal Strength

This involves measuring the strength of the arriving radio signal at the receiver. Knowing the power at which the signal was transmitted from the transmitter, the propagation characteristics of that particular radio signal in air and with information of the environment, it is possible to calculate approximately where the transmission originated based on how attenuated it is at the receiver. The unit is Received Signal Strength Indication (RSSI) which is a negative number represents the strength of the signal.

- Time measurement

In this method, the time it takes the radio signal to travel between transmitter and receiver is measured using one or more of a variety of different techniques and then knowing the speed of light, the distance can be calculated. This method can be executed using various concepts namely; Time of Arrival, Time of flight, Time Difference of Arrival, Angle of Arrival [17]. In this paper, concentration is focused on the time of arrival and time difference of arrival to achieve the objectives of the project because of their simplicity and accuracy.

➢ Time of arrival

Time of arrival is the simplest and most common ranging technique, most notable used in the Global Positioning System (GPS). This method is based on knowing the exact time that a signal was sent from the target, the exact time the signal arrives at a reference point, and the speed at which the signal travels [18]. Once these are known, the distance from the reference point can be calculated using the sample equation;

$$d = c * (t_{arrival} - t_{sent})$$

Where c is the speed of light. Using the distance, the set of possible location of the target can be determined. In two dimensions, this yields a circle with the equation below;

$$d = \sqrt{\left(x_{ref} - x\right)^2 + \left(y_{ref} - y\right)^2}$$

Where (x_{ref}, y_{ref}) is the known position of the reference point Once this set is calculated for enough reference points (atleast three for two dimensional or atleast four for three-dimensional), the exact position of the target can be calculated by finding the intersection.

➢ Time Difference of Arrival (TDOA)

Time Difference of Arrival is more versatile than Time of Arrival (TOA). This method does not require the time the signal was sent from the target, only the time the signal was received and the speed that the signal travels. Once the signal is received at two reference points, the difference in arrival time can be used to calculate the difference in distances between the target and the two reference points [18]. This difference can be calculated using the equation.

$$\Delta d = c * \Delta t$$

Where c is the speed of light and Δt is the difference in arrival times at each reference point. In two dimensions, this leads to the following equation.

$$\Delta d = \sqrt{(x_2 - x)^2 - (y_2 - y)^2} - \sqrt{(x_1 - x)^2 - (y_1 - y)^2}$$

Where (x_1, y_1) and (x_2, y_2) are the known positions of the base stations. Using nonlinear regression, this equation can be converted to the form of a hyperbola. Once the enough hyperbolas have been calculated, the position of the target can be calculated by finding the intersection.

Line of Sight Problem

Both TOA and TDOA measurements assume that the signal travels from the target to the reference point along the shortest, or the line of sight, path. When this path is blocked, the signal will arrive along a longer path, which results in a later time and incorrect target location. This problem is prevalent in navigation systems. For example, GPS systems lose signal in tunnels and dense urban areas, while indoor position systems often lose signal due to walls, furniture, or people. Several solutions have been proposed for this problem, including modeling the Non-Line of light path and using low-interference signals like Ultra-Wideband [18].

Trilateration, Triangulation and Multilateration are geometry methods used to determine the current location of the bus using either Radio signal strength or time measurements. Multilateration is the process of determining locations of points by measurement of distances from known points. Multilateration uses the time difference of Arrival of signals; the receivers only recognize the exact arrival time of the transmission signal. It is the only the runtime difference in comparison among the leading edge of the signals in three or more receivers on different sites. Triangulation is the process of determining the location of a point by measuring only angles to it from known points at either end of a fixed baseline, rather than measuring distances to the point directly as in trilateration. In this project, trilateration is used over other methods because of its simplicity.

2.3.3 Trilateration

The bus moves continuously which makes its current coordinates to change frequently. Trilateration, a geometry method will be used to determine the location of the bus as it moves along its way. Trilateration is the process of determining absolute or relative locations of points by measurement of distances, using the geometry of the environment. Trilateration has practical applications in surveying and navigation, including Global Positioning Systems. Every GPS satellite transmits information about its position and the current time at regular intervals. These signals are intercepted by a GPS receiver, which calculates how far away each satellite is based on how long it took for the message to arrive. GPS receivers take this information and use trilateration to calculate the user’s location [19]. Trilateration is used as the GPS receiver constantly receives and analyzes radio signals from multiple satellites and applies geometric circles (2 dimensions), spheres (3 dimensions) [20]. The point of intersection of the three circles, is the point of location of the object as shown in figure 5.

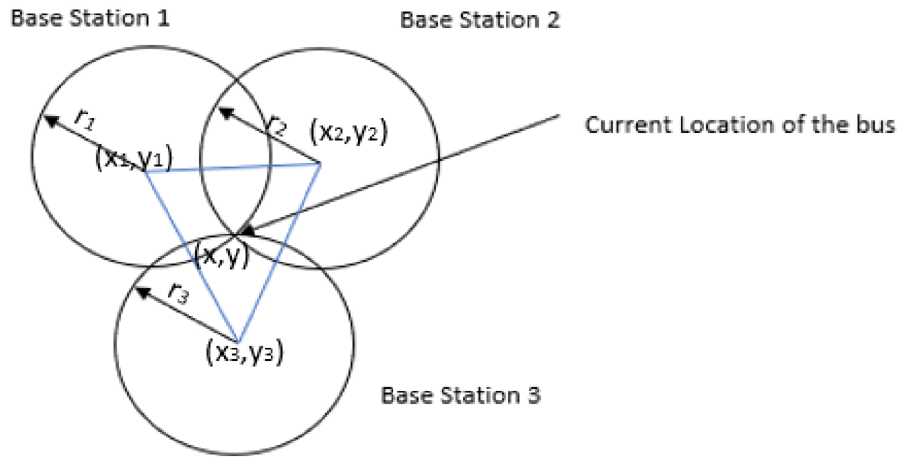

Figure 5: illustration showing the coordinates of the bus and base stations within its radius

The figure above shows the bus has (x,y) coordinates and the base station 1, 2 and 3 have (x_1,y_1), (x_2,y_2) and (x_3,y_3) respectively. Trilateration requires clock synchronization of the receiver and transmitter to be able to determine the time of arrival of the signal. The time of arrival of the signal is multiplied with the speed of light to get the distance covered by the signal (it is assumed that the LoRa signals travel at the same speed since it is also an electromagnetic wave). Using the time of arrival, the time taken by the signal to travel from the base station to the bus with a known velocity can be used to calculate the distance.

2.3.4 Time Of Arrival (TOA) Trilateration Algorithms model

In the (Time Of Arrival) TOA trilateration method, the location of the bus is determined by a single intersection point of three circles with center coordinates of three Base stations (1,2 and 3) as shown in figure 5 above. Assuming that (x, y) is the current coordinate of the bus and (x_i, y_i) is the i^{th} base station coordinates, the distance between the bus and the i^{th} base station is given by;

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}, \qquad i = 1,2,3 \ldots. \qquad (1)$$

Consider a LoRa transmitter on a bus travelling while transmitting signals to three different base stations, fixed points with antennas that receive and transmit signals in the cellular network [21]. Each base station having a maximum distance of 15km away from the bus to keep in range of the LoRa signals. The LoRa transceiver sends signals from the bus and based on the distances from the individual signals received by each LoRa receiver, a circle can be drawn from each base station to determine the intersection as shown in equation 1. The intersection will be the current location of the bus as shown in figure 1. This also implies that the LoRa transmitter needs to frequently send signals to the base stations to be able to track the bus in real time. Equation (1) is the conventional Time Of Arrival (TOA) trilateration method for finding the intersection of the circles. However, sometimes due to errors, the circles may not intersect at one point which creates three points of intersection as stated earlier. This creates need for a better algorithm which can put this into consideration and determine the location of the bus even when there are distance variations that can cause the circles not to intersect at one common point. The shortest distance algorithm explained below can be used to overcome this problem.

2.3.5 Shortest Distance algorithm

Consider two base stations as shown in the figure 6 below.

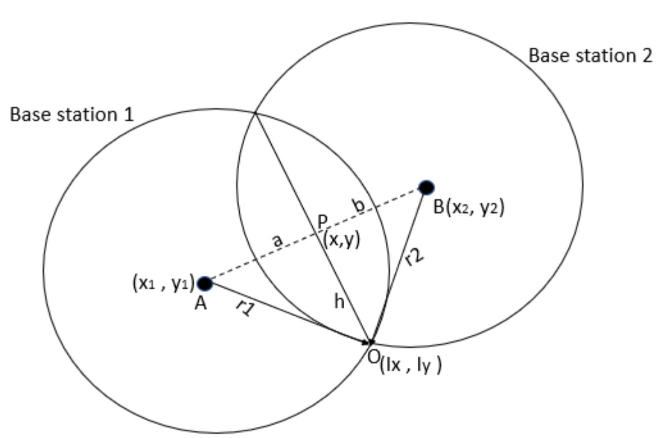

Figure 6: shows intersection of two circles from the radii of two base stations

The distance between points A and B can be given as;

$$d = a + b \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Distance d can also be expressed as;

$$d = \sqrt{\Delta x^2 + \Delta y^2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where $\Delta x = x_1 - x_2$ and $\Delta y = y_1 - y_2$.

Consider the two triangles AOP and BOP drawn from the centers of circles to join at point of intersection. The radii r1 and r2 can be given as;

$$r_1^2 = a^2 + h^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$r_2^2 = b^2 + h^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

The radii are obtained by multiplying the time of arrival of the LoRa signals with the speed of light since LoRa are electromagnetic waves as light waves.

Equating (3) and (4) and substituting (2) in the result gives the following result;

$$a = \frac{r_1^2 - r_2^2 + d^2}{2d} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Substituting (5) into (3), h can be expressed as;

$$h = \sqrt{r_1^2 - \frac{\left(r_1^2 - r_2^2 + d^2\right)^2}{4d^2}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

The possible two intersection points of the base stations (Ix, Iy) can be given by the following equations (7) and (8);

$$I_X = x \pm \frac{h(y_2 - y_1)}{d} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

$$I_y = y \pm \frac{h(x_2 - x_1)}{d} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (8)$$

Including the third base station, the same procedure is followed until other coordinates of intersection points corresponding (Base station 2, Base station 3) and (Base station 3 and base station 1) as shown in figure 7 below.

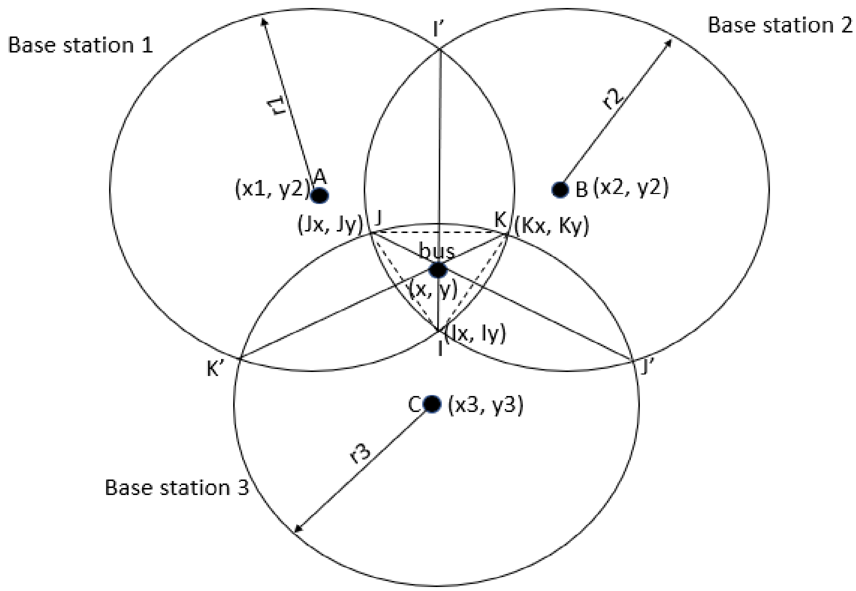

Figure 7: shows the determination of the position of the bus using the shortest distance algorithm

Finally, the average of the three selected interior intersection coordinates for the bus is the centroid of the triangle JKL calculated by;

$$X = \frac{(I_X + J_X + K_X)}{3} \quad , Y = \frac{(I_Y + J_Y + K_Y)}{3} \qquad \mathbf{(9)}$$

The shortest distance algorithm is simple and has generally good estimation performance, it determines the location of the bus to the averaged three intersection points.

The algorithms will be deployed on the web server;

- The code will be implemented on the local web server (Apache), an open source free software which offers various local host/servers on a computer [22].

The information used in the algorithm like coordinates, bus identification number, among others is also stored in an online database. This information can be accessed by the service providers for future improvement of their services.

2.4 Related work

Several papers have been written proposing the different systems that can be built to help travelers track their buses in real-time. The papers below describe related work in relation to the objectives of this project.

The paper [9] focuses on designing a low-power consumption system for geo-positioning without using GPS and GSM. LoRa technology is used because of its low power consumption which will enable battery life of the devices to last longer. Using LoRa and geometry methods is much cheaper compared to using GPS modules for tracking. The paper [9] also shows that there is no need of having GSM modules on each bus because they are replaced by LoRa modules, this also helps to cut costs since only one GSM module is used as the center point to transfer data to the web server. The paper [9] proposes a system that can be used by city buses to provide real time information to travelers like the current location of the bus. The proposed system consists of an end node, four gateways, server and application. The end node sends LoRa signals to the gateways using LoRaWAN. The gateways forward the packets via UDP/IP (User Datagram Protocol), a communication protocol used across the internet for mostly time-sensitive transmissions [23]. The time of arrival of the LoRa signals is then sent to The Things Network (TTN), an online open network that is used for IoT applications [24]. The TTN containing a Multilateration algorithm processes the data and routes the message to an application using the Message Query Telemetry Transport (MQTT) client, a machine-to-machine/IoT connectivity protocol that is lightweight and optimized to connect physical world devices with servers [25]. Finally, a third-party web application created using JAVA is used to process and analyze the data from the TTN to estimate the position.

Another paper [26] describes how college students and staff members can be assisted to know the current location of the bus to avoid being late for classes while waiting for buses at the station. To solve this challenge, a message will be displayed on the web application that will provide real-time information about the bus showing its time of arrival to reduce anxiety. The project [26] is implemented by installing GPRS modules on college buses which will transmit the current location from the GPS receiver to the server. The application then retrieves the data and stores it in a database from where the system displays real-time information of the bus [26]. However, [26] is an expensive approach to determining the location of the project. We seek to cut down costs by using LoRa transceivers and geometry methods to determine the location and then use only one GSM module to transmit the information to a web server other than installing GPS and GSM modules on each bus which is quite expensive.

In [17], The proposed tracking system consists of three modules; Bus unit, Central Control Unit and Client-Side Application. The Bus unit contains a GPS device that sends its coordinates after every fixed interval of time to the main server. GPS receiver collects signals from atleast three satellites to determine the location. The bus unit contains coordinates with a timestamp which is then compared with the previous coordinates and if there is any distinction then the coordinates are updated and sent to the server over GPRS network. The location details are stored on the server such as Identification (ID), longitude, latitude, timestamp, among others. Different unique ID numbers are assigned to each bus to differentiate the various buses. The server acts as a central repository of the system where all the information is stored and maintained. The server is also the intermediate between the bus module and user module. These databases contain real-time information such as bus routes, actual arrival/ departure time and real-time location of the bus. The user side module is an interactive web-based application that services the various function of the system to remote users. The user side module takes two inputs; the source that indicates the current location of the remote user and the second input is the destination of the user. When the customer sends a request, the application fires a query to the server for accessing the data stored within the server database and provides a list of the available buses. The user is then supposed to choose the bus range to see the amount of the time he or she has to wait. This project [17] helps students to plan well in order to avoid wasting time at the station. However, it still uses GPS devices which consume a lot of power.

This project seeks to use LoRa to enable travelers at stations track the bus they are waiting for in real time. The bus will contain A LoRa transmitter while the base stations will contain the LoRa receivers to create a wireless connection. The LoRa signals from the bus will be used to determine the location of the bus using geometry methods. A GSM module connected to the Arduino will be used to send the LoRa signal's time of arrival from the transmitter at the bus to an online server where computations will be carried out. The online server will contain a code that computes the estimated time of arrival of the bus and time will be fetched to a web application on screens at the stations as shown in figure 8 below.

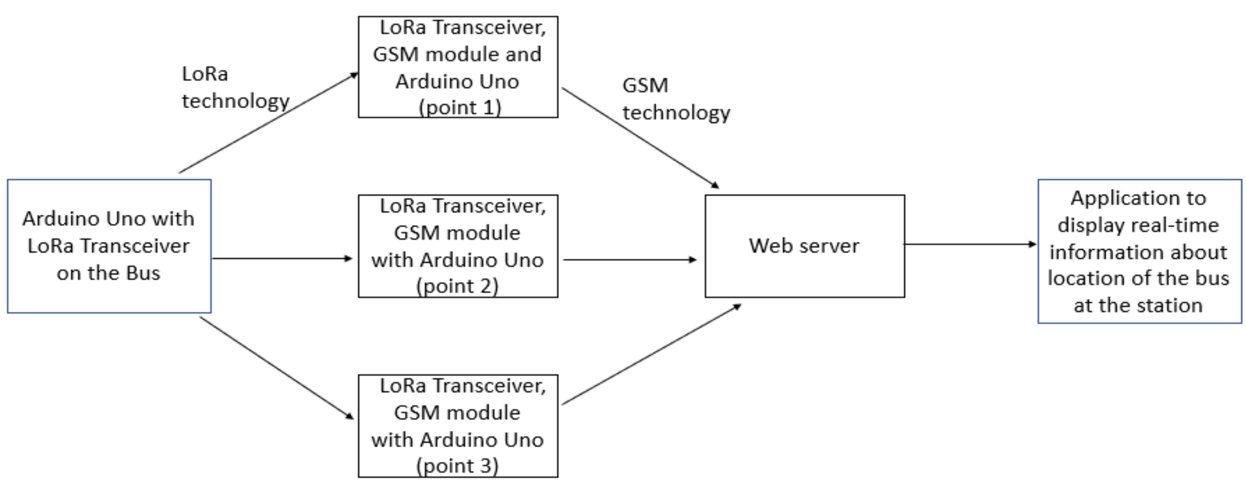

Figure 8: shows the different components of the Project

3. Design and methodology

The project consists of a LoRa transceiver installed in the bus. The LoRa transceiver is controlled by an Arduino Uno which sends out regular signals. Three Arduino Uno's each with a LoRa transceiver are installed in different places (stations) within a distance of 10km's from the bus. These LoRa transceivers receive the signal from the bus and GSM technology is used to send these signals further to a web server. The time of arrival of the signal from the bus to the stations is sent from the synchronized stations to the web server. Data is fetched from the server to the application to be displayed for the users.

The Pugh chart below shows the factors that were considered in selecting the microcontroller to be used in controlling the LoRa transceivers used in this project. The ranking is based on the selection criteria and each is graded out of five. One microcontroller having a less value in a certain field than the other means its better at that field. The Arduino microcontroller stood out and was selected for this project.

Table 2:shows the factors considered in selecting components of the project

Selection criteria	Arduino	Raspberry Pi	Arm Cortex-M
Cost	2	4	4
Programming skills needed	2	3	3
Size	2	2	2
Complexity	2	4	3
Power consumption	1	2	2
Total	9	15	14

The chart below shows the criteria followed in selecting the LoRa modules to be used in the project. The chart above describes the different LoRa modules with their respective qualities. The Murata-made LoRa module stood out because it has more interfaces to work with. This gives the project flexibility to try out the best communication interfaces and select the most suitable one that will be used to transmit the signals from the bus to the station.

Table 3: shows a comparison of different LoRa modules and their specifications

LoRa Module	Interfaces and Modulation	Supported frequencies	Price and extra features
Murata	I2C, UART, USB, SPI Integrated 32MHz clock (TCXO with frequency error=±2 ppm) and 32.768KHz clock (frequency error=±20 ppm)	868, 915 MHz, bit rate =4.8kb/s VDD = 3.3V	**Sold in bulk by Murata**
RN2483	UART FSK, GFSK and LoRa modulation	433 AND 868 MHz frequency bands VDD = 2.1 – 3.6V	**$20**
Zenseio	UART LoRa, FSK, GFSK	902 – 928 MHz, Data rate = 250bps to 12.5kbps VDD – 3.3V	**$25**
F8L10 series	UART	410 – 441 MHz, data bits = 8 VDD – 3.3, 5 V	**$18**

However, none of the above LoRa modules was readily available hence looking for alternatives. LoRa modules operate either at 434 or 918 MHz depending on the country communication policies. The LoRa module used is an RFM69HCW module which operates at frequency of 434MHz. This is an inexpensive and versatile radio module and needs a microcontroller to be able to send or receive signals [27]. 434 MHz is used because it is part of the unlicensed frequency bands in the Ghana which makes it quite easy for the project to be executed without violating the communications regulatory measures of the country.

Figure 9 below describes how the components are arranged.

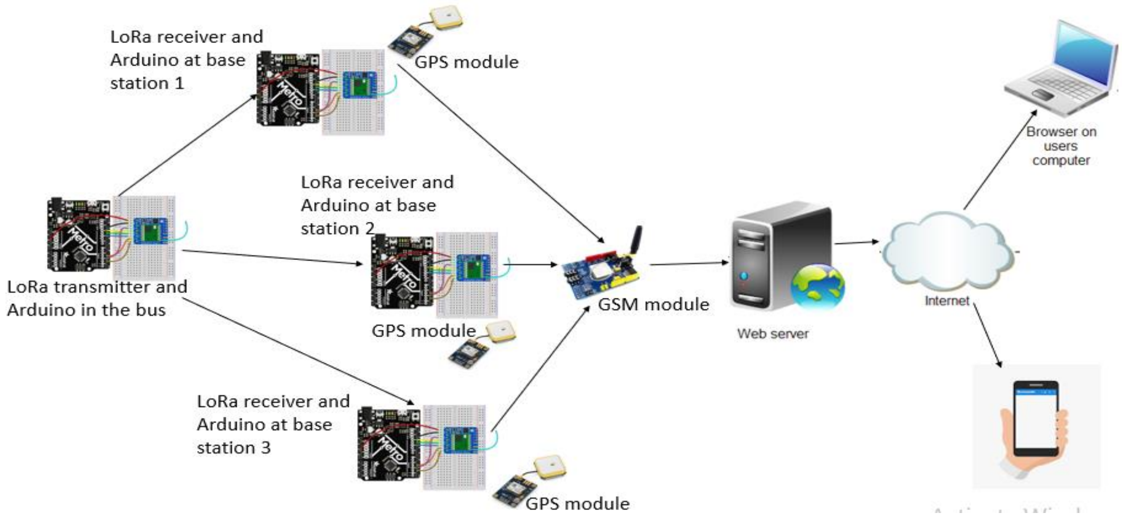

Figure 9:shows the flow of how the components are organized and coordinated

The structure contains four main parts namely; End node at the bus, three Arduinos which serve as gateways, Web server and Application platform as shown in figure 9 above. The end node is located in the bus and contains the LoRa transmitter and Arduino Uno, an open-source microcontroller development board [28], to send LoRa signals to the gateways. The end node sends out signals continuously at regular time intervals to be received by the gateways. For concurrent reception, a raspberry Pi should be used. The GPS module is used to indicate timestamps at which LoRa signals arrived at the base stations. The Arduino modules are also synchronized using these GPS timestamps.

Gateways at base stations

The three Arduino's with LoRa receivers at the base stations 1,2 and 3 serve as gateways because they the time of arrival of the LoRa signal to be sent over the internet to the web server with the help of a GSM module. The GSM module sends the signal's time of arrival from each base station to the server via the internet. The data received by the gateways is the time of arrival of the signal at a particular instance. The gateways are synchronized using the time from the GPS module and each time a signal arrives, the gateways record the signal's time of arrival and compare that to the current timestamp of a GPS Module. The different distances covered by the signal from bus to each gateway are sent to server. The server contains a trilateration code that computes for the current location of the bus using the provided distances.

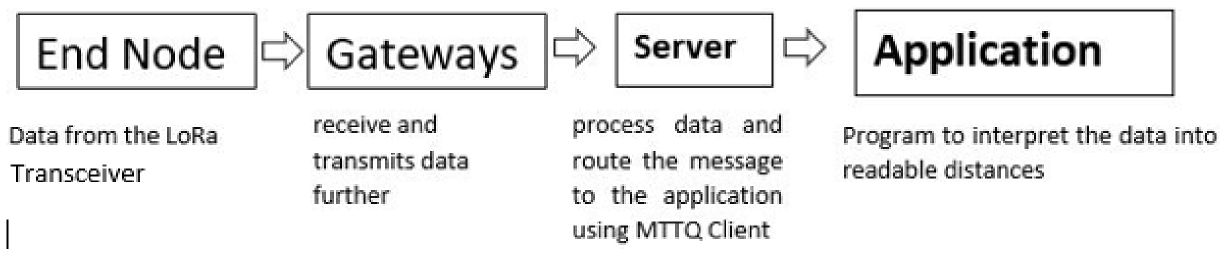

Figure 10:shows a simplified flow diagram of the project design

Web Server

A web server is a program that uses Hypertext Transfer Protocol (HTTP), a set of rules for transferring files on the World Wide Web [29], to serve the files that form web pages to users, in response to their requests. These are forwarded by their computers' HTTP clients [30]. In this project, it is responsible for decoding data from the Arduino Uno by using the trilateration algorithm installed on it and providing this information to the end user application. It is important that the three packets with the same payload but with different times travel arrive at the network server. The Web Server used in this project is a school engineering department server and access has been provided by the project supervisor.

User application

A software application that has to convert the data sent from the server to be readable by the traveler. The application consists mainly of two parts; map showing the current location of the bus and the estimated time of arrival. The main purpose is to obtain the data from the web server and display it at the bus stations' screen.

4. Results

The LoRa transmitter successfully transmitted LoRa signals to the receiver within a distance of 10m. The time difference of arrival was used rather than the time of arrival because it was easier to synchronize only the base stations as compared to time of arrival where the receiver and transmitter need to be synchronized. The GPS module recorded time in centiseconds which was not accurate enough to track significant changes in time as the transmitter moved. Synchronization was achieved by resetting all the Arduino boards at same time. The base stations where synchronized using interrupts in Arduino code because Arduino clocks can measure upto microseconds. The 'interrupts' method involved making one base station a 'master' while the rest of the base stations where 'slaves'; when the base stations are reset, the master serves as a control and all the slaves will have the same time as the master. Once the base stations are reset, the transmitter started sending signals and even though the base stations where in different locations, the base stations where synchronized. If the time of arrival of the LoRa signal to the different synchronized base stations is known, this gives an ideal of how far each base station is from the base station by using the time difference of arrival at the base stations.

However, this type of synchronization required the base stations to be in the same place about two meters apart which made it hard to acquire a variety of results. The table below shows the LoRa signals time of arrival recorded by the three base stations.

Transmitter

```
Feather RFM69 TX Test!
15:56:59.738 ->
RFM69 radio init OK!
15:56:59.874 -> RFM69 radio @434 MHz
Sending Hello World #0
15:57:00.870 -> Got a reply: And hello back to you
Sending Hello World #1
15:57:02.172 -> Got a reply: And hello back to you
Sending Hello World #2
Got a reply: And hello back to you
Sending Hello World #3
15:57:04.812 -> Got a reply: And hello back to you
```

Receiver (Base station)

```
Feather RFM69 RX Test!
15:59:43.593 ->
RFM69 radio init OK!
15:59:43.695 -> RFM69 radio @434 MHz
Received [16]: Hello World #125
15:59:45.139 -> RSSI: -80
15:59:45.139 -> Sent a reply
Time: 1813
Received [16]: Hello World #126
15:59:46.445 -> RSSI: -80
15:59:46.445 -> Sent a reply
Time: 3122
Received [16]: Hello World #127
15:59:47.759 -> RSSI: -80
15:59:47.759 -> Sent a reply
```

Figure 11: shows the results from the Transmitter to the receiver

Using Arduino code, the Transmitter first establishes connection with the Receiver and also checks whether they are at the same frequency. Then the LoRa transmitter will send a message inform of a message ‘Hello World’, the LoRa receiver will acknowledge recipient by showing ‘received Hello World’, the signal strength (RSSI) is recorded and a reply is sent back to the transmitter ‘hello back to you’, this continues as long there is a connection between the LoRa transmitter and receiver. The time taken for the LoRa signal to arrive at the receiver is recorded.

The TOA in microseconds for different positions of the transmitter from the base stations is displayed in the Arduino Serial monitor and recorded as shown in table 3 below.

Table 4: shows the results of the different locations of the Base Stations

		Test 1	Test 2	Test 3
BS	BS COORDINATES	TOA (μs)	TOA (μs)	TOA (μs)
1	(0.15,0.18)	228	815	1889
2	(1.05,1.18)	518	612	2105

3	(0.18,0.5)	789	1032	906

The figure below shows the setup of the base stations; the transmitter with two base stations close to it while the third base station is put in a far distance to get a variety of results

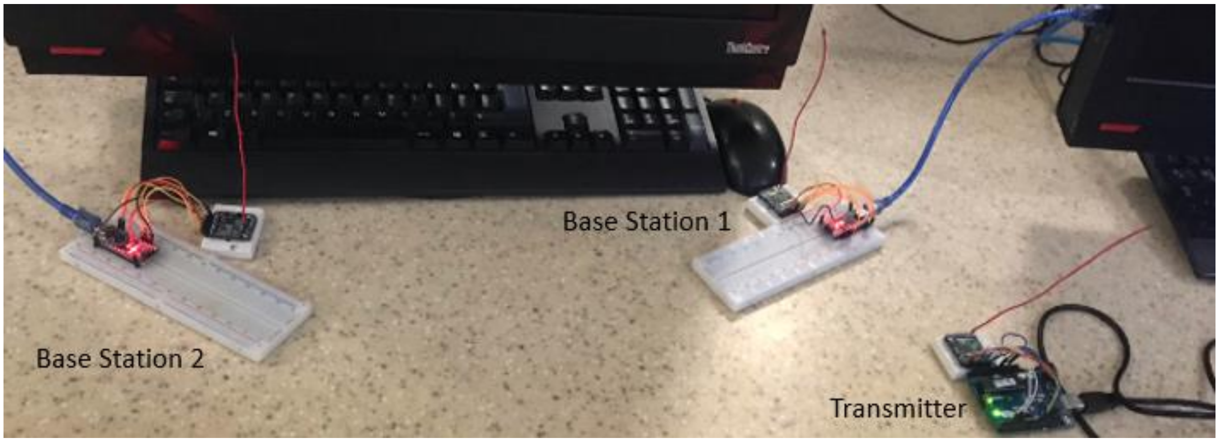

Figure 12: shows a picture of the Base Stations and the transmitter

The trilateration algorithm was implemented using Visual Studio Code and tested by feeding in various values of the base stations and time of arrival and it returned the value of the current location of the transmitter.

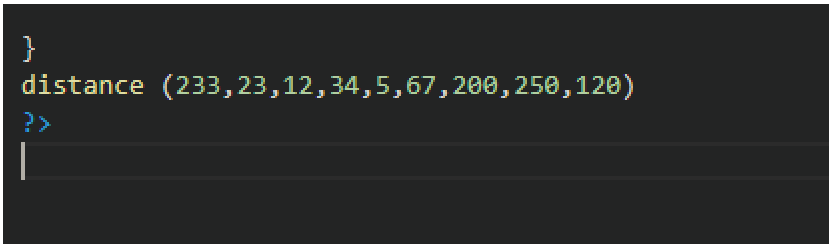

Figure 13: shows a screenshot of the code and the implementation of the algorithm

The screenshot above shows the function which takes in x and y co-ordinates of three base stations and the last values represent the different time of arrival at the base stations. The code was uploaded on Apache/ local host and then transferred to a web serve using FilleZilla, once the page was reloaded, it returned the value of the coordinates of the bus as (45,89).

4.2 Limitations

- LoRa has a low data rate and goes into sleep mode to save power which caused delays in signal arrival to the base station thus making it unsuitable for real-time applications.
- The LoRa signals where not able to travel far because of the desired antenna used to travel upto 15kms was not available.
- The accuracy of the GPS module used was not appropriate for this application because the time was reported in cent seconds which barely represented any change in the time of arrival of the LoRa signal from the receiver to all base stations.

5. Conclusion

This project suggested a low-cost and less power consuming approach of using LoRa technology, geometry methods to determine the location of the bus and only one GSM module for multiple buses to transmit the data to the internet. Results in the project show success in determining the location of the bus in real-time.

5.1 Future Work

The Real Time Clock (RTC) module should be used to synchronize the time of the base stations because it provides nanoseconds units of time which is appropriate for this project. The GPS module used in this project provided centiseconds which was did not provide any significant changes in time as the transmitter was moved in different positions. Appropriate linear antennas should be used with the LoRa modules, this will increase to coverage to a maximum of 15kms compared to the 10m achieved in this project. In this project, the scope was reduced to determining the location of the bus without developing a web application. However, in the future, a web application that shows the pinpoints of the bus in the city should be developed. A more robust trilateration algorithm should be used to consider the different scenarios of the different geometry inaccuracies like circles not interesting at one point, this will give room for inaccuracies to be catered for.

References

[1] M. M. M. J. W. Rabi G.Mishalani, "Passenger Wait Time Perceptions at Bus Stops: Empirical Results and Impact on Evaluating Real-Time Bus Arrival Information," *Journal of Public Transportation,* vol. 9, no. 2, 2006.

[2] S. L. K. Matiur Rahman, "Users' views on current and future real-time bus information systems," *Journal of Advanced Transportation ,* vol. 47, no. 3, 24 August 2012.

[3] T. H. N. H. K. O. Nobuaki Ohmori, "Passengers' Waiting Behavior at Bus Stops," *Research Gate ,* July 2008.

[4] M. Rouse, "Google Maps," February 2013. [Online]. Available: https://whatis.techtarget.com/definition/Google-Maps.

[5] T. Conneally, "Google Maps can now track buses and trains in real time," 2011. [Online]. Available: https://betanews.com/2011/06/08/google-maps-can-now-track-buses-and-trains-in-real-time/.

[6] K. Isabella, "How Does Google Maps Work and Gather Data?," 12 03 2019. [Online]. Available: http://www.itfixtech.com/blog/how-does-google-map-work-and-gather-data.

[7] J. C. Joseph Jenkins, "Ghana - Location, size and extent," April 2015 . [Online]. Available: https://www.nationsencyclopedia.com/Africa/Ghana-LOCATION-SIZE-AND-EXTENT.html.

[8] P. Goodman, "Wireless Network vs Wired Network: Advantages and Disadvantages," 23 April 2017. [Online]. Available: https://turbofuture.com/computers/Wireless-Network-vs-Wired-Network-Advantages-and-Disadvantages.

[9] M. N. P. Bernat Carbones Fargas, "GPS- free Geolocation using LoRa in Low-Power WANs," *In Proceedings of 2017 Global Internet of Things Summit(GIoTS),* 2017.

[10] I. I. O. T. (IIT), "Enhanced Data Rates for GSM Evolution," 2002.

[11] B. Ray, "What is LoRa? A technical Breakdown," 26 June 2018. [Online]. Available: https://www.link-labs.com/blog/what-is-lora.

[12] "What is LoRa?," 2012. [Online]. Available: https://www.semtech.com/lora/what-is-lora.

[13] A. R. John Kyes, "What is GPS?," 26 October 2017. [Online]. Available: https://www.geotab.com/blog/what-is-gps/.

[14] R. Barry, "IoT connectivity comparsion (GSM vs LoRa vs Sigfox vs NB-Iot)," Polymorph, Jan 2016. [Online]. Available: https://www.polymorph.co.za/iot-connectivity-comparison-gsm-vs-lora-vs-sigfox-vs-nb-iot/#.

[15] J. Rajasthan, "GPS," 13 04 2017. [Online]. Available: http://www.agranii.com/service/gps-tracking.

[16] Elektor, "Dragino LoRa IoT Development Kit," 22 09 2018. [Online]. Available: https://www.elektor.com/dragino-lora-iot-development-kit-868-mhz.

[17] DECAWAVES, "REAL TIME LOCATION SYSTEMS," 2014. [Online]. Available: https://www.decawave.com/wp-content/uploads/2018/08/aps003_dw1000_rtls_introduction.pdf.

[18] B. O'Keefa, "Finding Location with Time of Arrival and Time Difference of Arrival Techniques," *ECE Senior Capstone Project,* 2017.

[19] J.-L. D. C. A. S. M. S. Prosenjit Bose, "Art Gallery Localization," *NSERC,* 2018.

[20] F. Zahradnik, "What is Trilateration? How GPS devices use mathematics to determine positioning," 26 November 2018. [Online]. Available: https://www.lifewire.com/trilateration-in-gps-1683341.

[21] M. Rouse, "Base Station," March 2014. [Online]. Available: https://whatis.techtarget.com/definition/base-station.

[22] P. Ganesan, "What is Xampp?," 16 September 2017. [Online]. Available: https://www.wpblogx.com/what-is-xampp/.

[23] M. Rouse, "UDP (User Datagram Protocol)," *TechTarget Network,* June 2018.

[24] W. Giezman, "The things Network: Builidng a global IoT network in 6 months," 12 Jan 2016. [Online]. Available: https://medium.com/@wienke/the-things-network-building-a-global-iot-data-network-in-6-months-adc2c0b1ae9b.

[25] R. Gupta, "5 things to know about MQTT - The Protocol for Internet of Things," 23 September 2014. [Online]. Available: https://www.ibm.com/developerworks/community/blogs/5things/entry/5_things_to_know_about_mqtt_the_protocol_for_internet_of_things?lang=en.

[26] R. S. V. M. R. P. Dhruv Patel, "Real-Time Bus Tracking System," *International Research Journal of Engineering and Technology (IRJET),* vol. 04, no. 03, March 2017.

[27] M. Grusin, "RFM69HCW Hookup Guide," 31 May 2016. [Online]. Available: https://learn.sparkfun.com/tutorials/rfm69hcw-hookup-guide?_ga=1.268515319.767489989.1458921423.

[28] R. Sarafan, "Intro ro Arduino," 2015. [Online]. Available: https://www.instructables.com/id/Intro-to-Arduino/ .

[29] M. Rouse, "HTTP (HyperText Transfer Protocol)," July 2006. [Online]. Available: https://searchwindevelopment.techtarget.com/definition/HTTP.

[30] J. B. L. Margaret Rouse, "Web Server," July 2015. [Online]. Available: https://whatis.techtarget.com/definition/Web-server.

[31] B. Mitchell, "Learn the Meaning of a Network Gateway," 12 December 2018. [Online]. Available: https://www.lifewire.com/definition-of-gateway-817891.

[32] M. N. P. Bernat Carbones Fargas, "GPS-free Geolocation using LoRa in Low-Power WANs," *In Proceedings of 2017 Global Internet of Things Summit (GIoTs),* 2017.

[33] [Online].

[34] M. Rouse, "GPRS(General Packet Radio Services)," May 2007. [Online]. Available: https://searchmobilecomputing.techtarget.com/definition/GPRS.

[35] N. Bolton, "What are the advantages of GPRS?," 2018. [Online]. Available: https://www.techwalla.com/articles/what-are-the-advantages-of-gprs.

[36] M. Rouse, "GSM (Global System for Mobile communication)," September 2018. [Online]. Available: https://searchmobilecomputing.techtarget.com/definition/GSM.

[37] D. C.Wheeler, "The advantages and Disadvantages of GSM," 2017. [Online]. Available: https://www.techwalla.com/articles/the-advantages-and-disadvantages-of-gsm.

[38] T. Mwenje, "What is GSM ?," September 2015. [Online]. Available: https://www.technomag.co.zw/2015/09/17/what-is-gsm/.

[39] T. Magar, "GSM and GPRS: What exactly these are ? Difference between both," 25 October 2017. [Online]. Available: https://iotdunia.com/gsm-and-gprs/.

[40] J. Pillou, "What is WiFi and How does it work?," 2016. [Online]. Available: https://ccm.net/faq/298-what-is-wifi-and-how-does-it-work.

[41] S. Categoria, "Advantages and Disdvantages of WiFi or wireless networks," 2016. [Online]. Available: https://ventajasydesventajas.info/en/networks-wifi-wireless/.

[42] M. Patkar, "9 Things you wanted to know about Raspberry Pi," 12 August 2016. [Online]. Available: https://www.makeuseof.com/tag/9-things-wanted-know-raspberry-pi/.

[43] J.-y. P. G.-R. K. S. S. S.-s. H. Sajina Pradhan, "Enhanced location detection algorithms based on time of arrival trilateration," *IEEE Xplore Digital Library,* 2014.

[44] M. Rouse, "EDGE (Enhanced Data GSM Environment)," May 2007. [Online]. Available: https://searchmobilecomputing.techtarget.com/definition/EDGE.

[45] P. J. I. Paschal A. Ochang, "Evolutionary Analysis of GSM, UMTS and LTE Mobile Network Architectures," *World Scientific Nwes,* no. 54, 2016.

[46] Ericsson, "Technical Overview and performance of HSPA and Mobile WiMAX," Global mobile Suppliers Association , 2007.

[47] P. ,. T. G. M. Walker, "Base Stations and wireless networks: exposures and health consequences," in *Proceedings international Workshop on Base stations and Wireless Networks and Health Consequences* , June 15 - 16, 2005.

Appendix

Code for Transmitter

```
#include <SPI.h>
#include <RH_RF69.h>
/************ Radio Setup ********
// Change to 434.0 or other frequency, must match RX's freq!
#define RF69_FREQ 434.0
#if defined (__AVR_ATmega328P__)  // Feather 328P w/wing
  #define RFM69_INT     3  //
  #define RFM69_CS      4  //
  #define RFM69_RST     2  // "A"
  #define LED           13
#endif
// Singleton instance of the radio driver
RH_RF69 rf69(RFM69_CS, RFM69_INT);
int16_t packetnum = 0;  // packet counter, we increment per xmission
void setup()
{
  Serial.begin(115200);
  //while (!Serial) { delay(1); } // wait until serial console is open, remove if not tethered to computer
   pinMode(LED, OUTPUT);
  pinMode(RFM69_RST, OUTPUT);
  digitalWrite(RFM69_RST, LOW);
```

```
  Serial.println("Feather RFM69 TX Test!");
  Serial.println();
  // manual reset
  digitalWrite(RFM69_RST, HIGH);
  delay(10);
  digitalWrite(RFM69_RST, LOW);
  delay(10);

  if (!rf69.init()) {
    Serial.println("RFM69 radio init failed");
    while (1);
  }
  Serial.println("RFM69 radio init OK!");
  // Defaults after init are 434.0MHz, modulation GFSK_Rb250Fd250, +13dbM (for low power module)
  // No encryption
  if (!rf69.setFrequency(RF69_FREQ)) {
    Serial.println("setFrequency failed");
  }
  // If you are using a high power RF69 eg RFM69HW, you *must* set a Tx power with the
  // ishighpowermodule flag set like this:
  rf69.setTxPower(20, true);  // range from 14-20 for power, 2nd arg must be true for 69HCW
  // The encryption key has to be the same as the one in the server
  uint8_t key[] = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08,
                    0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08};
  rf69.setEncryptionKey(key);
    pinMode(LED, OUTPUT);
  Serial.print("RFM69 radio @");  Serial.print((int)RF69_FREQ);  Serial.println(" MHz");
}
void loop() {
  delay(1000);  // Wait 1 second between transmits, could also 'sleep' here!
```

```
  char radiopacket[20] = "Hello World #";
  itoa(packetnum++, radiopacket+13, 10);
  Serial.print("Sending "); Serial.println(radiopacket);

  // Send a message!
  rf69.send((uint8_t *)radiopacket, strlen(radiopacket));
  rf69.waitPacketSent();

  // Now wait for a reply
  uint8_t buf[RH_RF69_MAX_MESSAGE_LEN];
  uint8_t len = sizeof(buf);
  if (rf69.waitAvailableTimeout(500))  {
    // Should be a reply message for us now
    if (rf69.recv(buf, &len)) {
      Serial.print("Got a reply: ");
      Serial.println((char*)buf);
      Blink(LED, 50, 3); //blink LED 3 times, 50ms between blinks
    } else {
      Serial.println("Receive failed");
    }
  } else {
    Serial.println("No reply, is another RFM69 listening?");
  }
}
void Blink(byte PIN, byte DELAY_MS, byte loops) {
  for (byte i=0; i<loops; i++)  {
    digitalWrite(PIN,HIGH);
    delay(DELAY_MS);
    digitalWrite(PIN,LOW);
    delay(DELAY_MS);
```

```
  }
}
```

Code for Receivers

```
#include <SPI.h>
#include <RH_RF69.h>
/************ Radio Setup ***************/

// Change to 434.0 or other frequency, must match RX's freq!
#define RF69_FREQ 434.0
unsigned long time;
#if defined (__AVR_ATmega328P__)  // Feather 328P w/wing
  #define RFM69_INT     3  //
  #define RFM69_CS      4  //
  #define RFM69_RST     2  // "A"
  #define LED           13
#endif

// Singleton instance of the radio driver
RH_RF69 rf69(RFM69_CS, RFM69_INT);
int16_t packetnum = 0;  // packet counter, we increment per xmission
void setup()
{
  Serial.begin(115200);
  //while (!Serial) { delay(1); } // wait until serial console is open, remove if not tethered to computer
  pinMode(LED, OUTPUT);
  pinMode(RFM69_RST, OUTPUT);
  digitalWrite(RFM69_RST, LOW);
  Serial.println("Feather RFM69 RX Test!");
```

```
  Serial.println();
  // manual reset
  digitalWrite(RFM69_RST, HIGH);
  delay(10);
  digitalWrite(RFM69_RST, LOW);
  delay(10);

  if (!rf69.init()) {
    Serial.println("RFM69 radio init failed");
    while (1);
  }
  Serial.println("RFM69 radio init OK!");
  // Defaults after init are 434.0MHz, modulation GFSK_Rb250Fd250, +13dbM (for low power module)
  // No encryption
  if (!rf69.setFrequency(RF69_FREQ)) {
    Serial.println("setFrequency failed");
  }
  // If you are using a high power RF69 eg RFM69HW, you *must* set a Tx power with the
  // ishighpowermodule flag set like this:
  rf69.setTxPower(20, true);  // range from 14-20 for power, 2nd arg must be true for 69HCW

  // The encryption key has to be the same as the one in the server
  uint8_t key[] = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08,
                    0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08};
  rf69.setEncryptionKey(key);

  pinMode(LED, OUTPUT);
```

```
  Serial.print("RFM69 radio @");  Serial.print((int)RF69_FREQ);  Serial.println(" MHz");
}
void loop() {
 if (rf69.available()) {
    // Should be a message for us now
    uint8_t buf[RH_RF69_MAX_MESSAGE_LEN];
    uint8_t len = sizeof(buf);
    if (rf69.recv(buf, &len)) {
      if (!len) return;
      buf[len] = 0;
      Serial.print("Received [");
      Serial.print(len);
      Serial.print("]: ");
      Serial.println((char*)buf);
      Serial.print("RSSI: ");
      Serial.println(rf69.lastRssi(), DEC);
      if (strstr((char *)buf, "Hello World")) {
        // Send a reply!
        uint8_t data[] = "And hello back to you(BS 1)";
        rf69.send(data, sizeof(data));
        rf69.waitPacketSent();
        Serial.println("Sent a reply (BS 1)");
        Blink(LED, 40, 3); //blink LED 3 times, 40ms between blinks
        Serial.print("Time: ");
        time = millis();
        Serial.println(time);
      }
    } else {
```

```
      Serial.println("Receive failed");
    }
  }
}



void Blink(byte PIN, byte DELAY_MS, byte loops) {
  for (byte i=0; i<loops; i++)  {
    digitalWrite(PIN,HIGH);
    delay(DELAY_MS);
    digitalWrite(PIN,LOW);
    delay(DELAY_MS);
  }
}
```

<u>Trilateration Code</u>

```
<?php

function distance
($mobstationx,$mobstationy,$bstation1x,$bstation1y,$bstation2x,$bstation2y,$bstat
ion3x,$bstation3y){
$f = 433 * pow(10,6) ;//MHZ
$c = 3* pow(10,8) ; //speed of light
  $d1 = sqrt(pow(($mobstationx - $bstation1x),2) + pow(($mobstationy -
$bstation1y),2));
  $d2 = sqrt(pow(($mobstationx - $bstation2x),2) + pow(($mobstationy -
$bstation2y),2));
  $d3 = sqrt(pow(($mobstationx - $bstation3x),2) + pow(($mobstationy -
$bstation3y),2));

  $m1 = ceil(($d1 * $f)/$c);
  $m2 = ceil(($d2 * $f)/$c);
  $m3 = ceil(($d3 * $f)/$c);

  $ed1 = (($m1 * $c)/$f);
  $ed2 = (($m2 * $c)/$f);
  $ed3 = (($m3 * $c)/$f);
```

```
  echo ("Distance from base station 1 $d1<br>"."distance from base station 2
$d2<br>"."distance from base station 3 $d3<br>");

//////////////////////////////basestation 1 and 2
$delta_x = $bstation2x - $bstation1x;
$delta_y = $bstation2y - $bstation1y;
$delta = sqrt(pow(($delta_x),2) + pow(($delta_y),2));
$k = (pow(($delta),2)+ pow($ed1,2) - pow($ed2,2))/(2*$delta);

$Ix = $bstation1x + (($delta_x*$k)/$delta) -
(($delta_y/$delta)*(sqrt(pow($ed1,2)-pow($k,2))));
$Iy = $bstation1y + (($delta_y*$k)/$delta) +
(($delta_x/$delta)*(sqrt(pow($ed1,2)-pow($k,2))));

$Ixp = $bstation1x + (($delta_x*$k)/$delta) +
(($delta_y/$delta)*(sqrt(pow($ed1,2)-pow($k,2))));
$Iyp = $bstation1y + (($delta_y*$k)/$delta) -
(($delta_x/$delta)*(sqrt(pow($ed1,2)-pow($k,2))));

//////////////////////////////// base station 2 and 3
$delta_xj = $bstation3x - $bstation2x;
$delta_yj = $bstation3y - $bstation2y;
$deltaj = sqrt(pow($delta_xj,2) + pow($delta_yj,2));
$k1 = (pow($deltaj,2) + pow($ed2,2) - pow($ed3,2))/(2*$deltaj);

$jx = $bstation2x + (($delta_xj*$k1)/$deltaj) -
(($delta_yj/$deltaj)*(sqrt(pow($ed2,2)-pow($k1,2))));
$jy = $bstation2y + (($delta_yj*$k1)/$deltaj) +
(($delta_xj/$deltaj)*(sqrt(pow($ed2,2)-pow($k1,2))));

$jxp = $bstation2x + (($delta_xj*$k1)/$deltaj) +
(($delta_yj/$deltaj)*(sqrt(pow($ed2,2)-pow($k1,2))));
$jyp = $bstation2y + (($delta_yj*$k1)/$deltaj) -
(($delta_xj/$deltaj)*(sqrt(pow($ed2,2)-pow($k1,2))));

/////////////////////////////base station 1 and 3
$delta_xk = $bstation3x - $bstation1x;
$delta_yk = $bstation3y - $bstation1y;
$deltak = sqrt(pow(($delta_xk),2) + pow(($delta_yk),2));
$k2 = (pow($deltak,2)+ pow($ed1,2) - pow($ed3,2))/(2*$deltak);

$kx = $bstation1x + (($delta_xk*$k2)/$deltak) -
(($delta_yk/$deltak)*(sqrt(pow($ed1,2)-pow($k2,2))));
$ky = $bstation1y + (($delta_yk*$k2)/$deltak) +
(($delta_xk/$deltak)*(sqrt(pow($ed1,2)-pow($k2,2))));
```

```
$kxp = $bstation1x + (($delta_xk*$k2)/$deltak) +
(($delta_yk/$deltak)*(sqrt(pow($ed1,2)-pow($k2,2))));
$kyp = $bstation1y + (($delta_yk*$k2)/$deltak) -
(($delta_xk/$deltak)*(sqrt(pow($ed1,2)-pow($k2,2))));

$x_tilde = ($Ix + $jx + $kx)/3;
$y_tilde = ($Iy + $jy + $ky)/3;

echo("($x_tilde".",$y_tilde)");

}
distance (233,23,12,34,5,67,200,250,120)
?>
```