GPS-free Geolocation using LoRa in Low-Power WANs

Fargas, Bernat Carbones; Petersen, Martin Nordal

Published in:
Proceedings of 2017 Global Internet of Things Summit (GIoTS)

Link to article, DOI:
10.1109/GIOTS.2017.8016251

Publication date:
2017

Document Version
Peer reviewed version

Citation (APA):
Fargas, B. C., & Petersen, M. N. (2017). GPS-free Geolocation using LoRa in Low-Power WANs. In Proceedings of 2017 Global Internet of Things Summit (GIoTS) [8016251] IEEE. https://doi.org/10.1109/GIOTS.2017.8016251

GPS-free Geolocation using LoRa in Low-Power WANs

Bernat Carbonés Fargas, Martin Nordal Petersen
Department of Photonics Engineering
Technical University of Denmark
2800 Kongens Lyngby, Denmark
bernat.carbones.fargas@gmail.com; mnpe@fotonik.dtu.dk

***Abstract*—Internet of Things (IoT) has been growing over the last few years in multiple applications and due to the high need GPS for tracking capabilities, an innovative opportunity arises. This paper reports on a design and implementation of a LoRaWAN tracking system which is capable exploiting transmitted packages to calculate the current position without the use of GPS or GSM. This is done using the low power technology LoRa where the geolocation is calculated applying a multilateration algorithm on the gateways timestamps from received packages. The whole system consisted of an end-node, four gateways, a server and a java application to store the obtained data in a MySQL database.**

Keywords—IoT, LoRaWAN, geolocation, TDOA

I. Introduction

Nowadays, elderly assisted living in outdoors localization is mainly done using Global Navigation Satellite Systems (GNSS) receivers [1]. These kind of receivers are also used in other application for tracking pets or wildlife animals. However, it is particularly important in relation to elderly people suffering from certain forms of dementia. This segment of elderly can occasionally get lost and need recovery if not guided back. Up until now GPS receivers combined with GSM network for transmission of coordinates have been used to face this problem. Currently, there are affordable GPS receivers available on the market, but their main problem is their battery lifetime and need to be recharged every few days. The current consumption of a GPS receiver is about 30-50 mA, which is considerable more energy required inform most low power IoT devices. For instance, a LoRaWAN module working at a 868 MHz band consumes 2.8 mA, in the “on” state, 38.9 mA transmitting data and 14.2 mA receiving data [2]. Therefore, the motivation of this work arises from the need of designing a low-power consumption system allowing geopositioning without using GPS and GSM. At this point, LoRa plays an important role due to its interesting features and also to the new innovative applications that are growing. A low-cost device enabling geopositioning with a long battery lifetime would be useful for securing successful recovery of people in need.

II. LoRaWAN for Geolocation

LoRaWAN presents interesting features, so it becomes a powerful technology in geolocation applications. For instance, the long range that can be reached, up to 15 km in rural areas and 5 km in urban areas due to the good sensitivity of the receivers (-130 dBm). Consequently, the total path of the link budget could reach up to 155 dB [3]. Another interesting characteristic is the bandwidth, which is bigger than other IoT technologies in LPWANs, so it is better to distinguish different paths from the same signal (useful for tracking capabilities in urban scenarios where reflections are present) [4]. Furthermore, various studies about LoRa technology for geolocation have been carried out in recent years, in which Sagemcom obtained good accuracy results (up to 4 meters). However, 42 gateways were used in a hexagonal layout to improve the results and the signal from the end-node was received by at least 10 gateways, when the data rate was at the highest [5]. Finally, LoRaWAN is an open source technology, there is information about its implementation, layers, packet structure, protocols of communications and other relevant features.

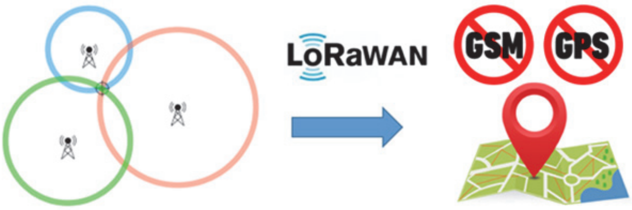


Fig. 1. LoRaWAN geolocation based on time of flight analysis of the packages sent from a low power IoT device. No GPS or GSM units were used.

It is in this context, important to distinguish the difference between LoRa and LoRaWAN. On one hand, LoRa is a modulation based on a variation of chirp spread spectrum (CSS) with integrated Forward Error Correction (FEC) [6]. Therefore, it uses the entire channel bandwidth to broadcast a signal, making it robust to channel noise and resistant to multipath, fading and the Doppler effect, even at low power. In the CSS modulation, 6 spreading factors (SF) are defined, from SF = 7 to SF = 12, that ensure orthogonal transmissions at different data rates. On the other hand, LoRaWAN is a media access control (MAC) protocol designed to allow low-powered devices to communicate with Internet connected applications over LPWAN. It is fully bidirectional and was architected by security experts to ensure reliability and safety [7]. LoRaWAN can be mapped to the second and third layer of the OSI model, on top of LoRa modulation. The LoRaWAN protocol is defined and standardized by the LoRa Alliance.

III. System Design

The overall system consisted mainly of four blocks: an end-node, four gateways, The Things Network (TTN) and a third-party application (Fig. 2). The end-node sent the data over the air using the LoRaWAN protocol and the gateways that were close enough (around 5 km) received the data. Then, the gateways forwarded the packets via UDP/IP to TTN, together with information from the received signal such as the exact time when the packet was received, the RSSI, the working frequency, etc. Afterwards, TTN processed the data from the different gateways and routed the messages to the application using a Message Queue Telemetry Transport (MQTT) client. Finally, the algorithm to estimate the position was applied in a third- party application.

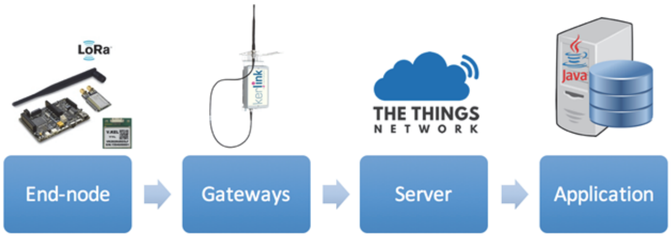


Fig. 2. Elements of the whole system

A. End-node

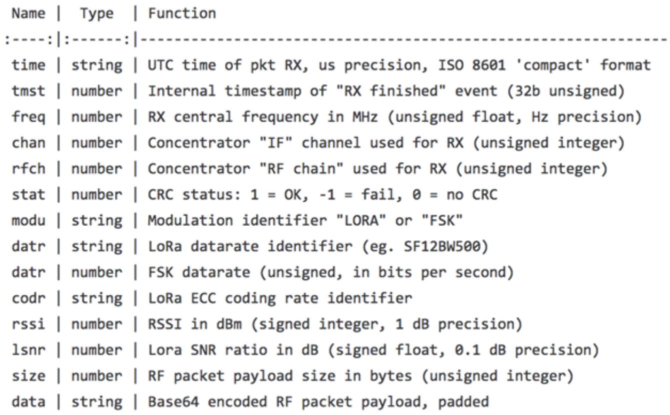
The end-node was responsible for sending the data acquired from a GPS receiver over the air through a LoRaWAN module. A GPS module was used to be able to benchmark and reference the performance of the investigated geolocation method. A Waspmote board was the main device of the end-node, so it processed and gathered information from the other modules. The Waspmote's core is based on the Atmel ATmega1281 microcontroller and is suitable for IoT applications. The modular architecture allows to integrate only the modules needed in each situation. Waspmote was selected as the core processor because of its low power consumption, which was one of the main requirements of the system. The feasibility for allowing to connect a LoRaWAN module was also relevant for choosing this device. The GPS coordinates were transmitted approximately every 10 seconds over the LoRaWAN and the Waspmote parsed the coordinates and transmitted them as a payload in the packet.

B. Gateways

The main function of the four gateways was to route the data received from the end-node to the server via UDP / IP. In order to estimate the location of the device, the received time of the packet from each gateway was needed to apply the multilateration algorithm. There are several gateways supporting LoRa technology on the market. The chosen one was Kerlink because a GPS receiver is embedded in the gateway. Therefore, all gateways are synchronized by using the timestamp from the GPS satellites.

The protocol between the gateway and the server is set in a binary file called "packet forwarder" that runs inside the gateway. There is no authentication of the gateway nor the server, and acknowledgements are only used for network quality assessment, not to correct lost packets. Some types of packets are exchanged between the gateway and the server. A complete description can be found in [8]. One of the packets exchanged contains the payload of the packet received from the end-node. This is transmitted in a JSON object called "rxpk" (Fig. 3). The "time" field contains the information which is relevant for the algorithm. Furthermore, other information about the received signal is forwarded to the server: the data rate, the received signal strength indicator (RSSI), the signal-to-noise ratio (SNR) or the coding rate.

At the beginning, the value set in the "time" field had microseconds accuracy, which translates to around 300 meters in distance. Consequently, in order to enhance the accuracy, the binary file was modified to have nanoseconds (0.3 meters). All raw C code had to be cross-compiled on a Linux system using an ARM toolchain provided by Kerlink. The C code that provides the binary file and the steps on how to do the cross compilation can be found in a GitHub repository [9].



```
 Name |  Type  | Function
:----:|:------:|--------------------------------------------------------------
 time | string | UTC time of pkt RX, us precision, ISO 8601 'compact' format
 tmst | number | Internal timestamp of "RX finished" event (32b unsigned)
 freq | number | RX central frequency in MHz (unsigned float, Hz precision)
 chan | number | Concentrator "IF" channel used for RX (unsigned integer)
 rfch | number | Concentrator "RF chain" used for RX (unsigned integer)
 stat | number | CRC status: 1 = OK, -1 = fail, 0 = no CRC
 modu | string | Modulation identifier "LORA" or "FSK"
 datr | string | LoRa datarate identifier (eg. SF12BW500)
 datr | number | FSK datarate (unsigned, in bits per second)
 codr | string | LoRa ECC coding rate identifier
 rssi | number | RSSI in dBm (signed integer, 1 dB precision)
 lsnr | number | Lora SNR ratio in dB (signed float, 0.1 dB precision)
 size | number | RF packet payload size in bytes (unsigned integer)
 data | string | Base64 encoded RF packet payload, padded
```

Fig. 3. rxpk JSON

C. The Things Network

TTN [10] was responsible for decoding data from the four gateways and transmitting it to the third-party application. It was important that the four packets from the different gateways, with the same payload but with different times, arrived at the network server. The most relevant feature for the algorithm computation was the value of the time field in the JSON which was sent. TTN is an open-source platform and allows a wide variety of third-parties to be connected. TTN offers a feature full and useful web interface to see the data that has been sent. However, if the goal is to process and analyze such information, a communication with a third-party application is required. Thus, a MQTT client was used to forward the data to a Java application.

D. Third-party application

The third-party application consisted mainly of two parts: a Java application and a MySQL database. The main function was to obtain the data from TTN, parse it and insert it in the database for processing in the following step. As already stated, a MQTT client was used to establish the communication with the server. The MQTT is a machine-to-machine (M2M) connectivity protocol which is designed as an extremely lightweight publish and subscribe messaging transport. TTN uses MQTT to publish device activations and messages, but also allows the user to publish a message for a specific device in response. Therefore, a subscription was done in the Java code to the desired topic with the suitable device identifier in order to obtain all packets from the gateways. The object-oriented Java code was composed by several classes to parse and manage the data from the server, as well as a JDBC driver to establish connection with the MySQL database.

IV. Triangulation in LoRaWAN networks

There are several techniques which can be used to estimate the position of the device, each one of them with different features. It is important to select the most suitable one depending on the known information from the end-node. The three most common methods used for performing the geolocation are triangulation, trilateration and multilateration. Triangulation uses angles of incidence of the signal received from the transmitter. A triangle is defined with two of them and the end-node position is estimated applying trigonometric formulas. Trilateration requires the distance between the transmitter and the receiver, which can be obtained from the time of arrival (TOA), the time of flight (TOF) or from the received signal strength indicator (RSSI). Therefore, it requires synchronization between the transmitter and the receiver. The position is the intersection of the three circles obtained from the different distances. Multilateration is quite similar to trilateration; however, the main feature to compute the location is the time difference of arrival (TDOA). The transmitters are synchronized to each other, whereas the receiver does not need to be. Thus, the location in this technique is the intersection of at least two hyperbolas (three antennas required). The tracking IoT system did not have synchronization with the end-node, only the gateways were synchronized with each other. Therefore, the information available was the time when the packet was received by each gateway. The TDOA was computed with this information and, for this reason, the multilateration algorithm was chosen. The RSSI was also known, so trilateration could also be applied. However, recent studies demonstrate a better accuracy using TDOA instead of RSSI [5].

The system consisted of several phases, as shown in Fig. 4.

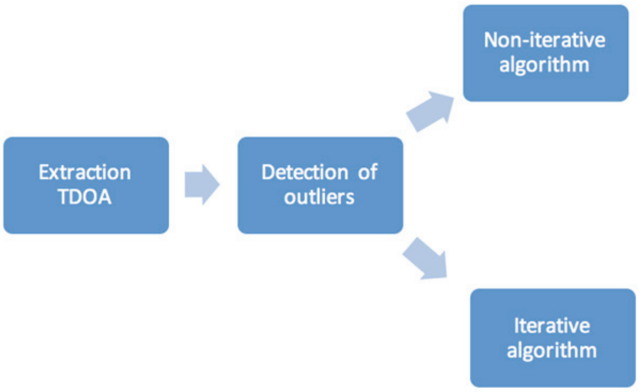


Fig. 4. Algorithm structure

A. Extraction of TDOA

The first step consisted in computing the TDOA based on a pair of UTC times. Therefore, once a packet was received from the end-node, four different UTC times (one from each gateway) were inserted into the database. Then, the TDOAs ($\mathrm{t_{ij}}$) were calculated as follows:

$$\mathrm{t_{ij}} = \mathrm{t_i} - \mathrm{t_j} \qquad \forall\, \mathrm{i,j} = 1{:}4 \quad \mathrm{j} \neq \mathrm{i} \tag{1}$$

Therefore, twelve TDOAs were computed, but only six of them are useful since one is the the opposite from the other.

B. Detection of outliers

According to Barnett and Lewis [11], an outlier is defined as "an observation which appears to be inconsistent with the remainder of that set of data". Keeping outliers in a dataset can lead to wrong results, so it is important to detect the true outliers. However, in some cases it may not be possible to determine if an outlying point is bad data. There are different methods to detect outliers. Grubbs' test, which detects one outlier at a time assuming a normal distribution. The Tietjen-Moore test is a generalization of the Grubbs' test to the case of multiple outliers. Nevertheless, it has a limitation: the number of outliers must be specified exactly. Finally, the Generalized Extreme Studentized Deviate (ESD) test is also a generalization of the Grubbs' Test to the case of more than one outlier, but it does not require to know the number of them. This test only requires only an upper bound for the suspected number of outliers. Given the upper bound *r*, the Generalized ESD test performs *r* separate tests: a test for one outlier, a test for two outliers and so on up to *r* outliers. As the number of outliers was unknown, the Generalized ESD was applied to detect them in the six TDOA cased mentioned above.

C. Non-iterative algorithm

There are several approaches to estimate the geolocation once the TDOAs has been obtained. In this work, two algorithms were applied: a *non-iterative* and an *iterative.*

The non-iterative algorithm was based on using a linear multilateration technique to estimate the position. In order to do so, the following inputs were needed: the TDOAs and also the locations of the four gateways.

The position over the globe can be expressed in Cartesian coordinates (x, y, z) or in Geodetic coordinates (latitude, longitude and height). The first set of coordinates is useful for mathematical calculations and easier to manipulate, but not for providing understandable information. The second one provides understandable information, but it is useless for mathematical calculations. Therefore, the known Geodetic coordinates from the gateways could not be used directly into the algorithm and a transformation was needed. The WGS-84 (REF) was the reference ellipsoid to perform such transformation since all the coordinates from the gateways are referred to this ellipsoid and it is the one used in GPS. Once the coordinates from the gateways were transformed to Cartesian (2D), the linear multilateration algorithm could be applied. The gateways coordinates were denoted as (x_i, y_i) and the unknown position of the end-device as (x, y). The scenario was the following (Fig. 5):

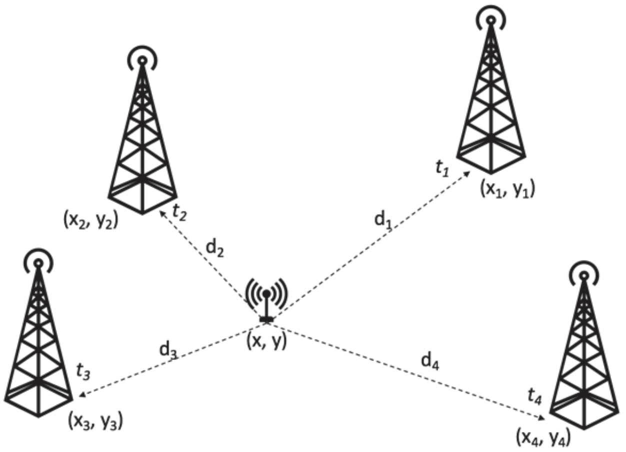


Fig. 5. Scenario of the algorithm

The multilateration algorithm is linear when 4 gateways are used because one extra unknown is added: the distance from one gateway to the end-node. Thus, in the case of 3 gateways, the computational resources needed are higher to solve the non-linear system of equations. Finally, the estimated position of the end-node (x, y) was transformed to Geodetic coordinates to compute the error and display it in a map.

D. Iterative algorithm

When using Cartesian coordinates as done in the method above, a possibility of making errors in the calculated distances arises. Hence, a new proposal was designed with the goal of always working with Geodetic coordinates. This new technique was based on the Haversine formula (REF) to compute the distance between two points over the globe. The first step was to create a grid of possible latitude and longitude values of the end-node. As the region where the movement of the end-node was known, the values were restricted to this area. The scenario was the same as in the non-iterative algorithm (Fig. 5), but the procedure to follow was different. For all possible latitude and longitude values, the distances from each of the four gateway to the end-node were computed. The three differential distances computed using the measured TDOAs were compared with the previous ones. Therefore, the pair of latitude and longitude values providing the minimum error were selected.

V. Tests and Results

The gateways were distributed around the area taking into account some aspects. First of all, they had to form a four-sided polygon, each side with a length of around 2 to 3 km. The location was also selected according to the altitude, since the higher the antenna, the larger the coverage. This improves the sensitivity of the device, because if the height of the antenna is doubled, a gain of 6 dB is achieved considering the flat Earth model. Therefore, the positions of the gateways were the following (Fig. 6):

- DTU (1): One gateway set permanently on the roof of the 344 building at The Technical University of Denmark.
- Virum (2): One gateway located at the top of a high building in Virum, Denmark.
- Bagsværd (3): Another one set on the roof of a house in Bagsværd, in front of Bagsværd lake.
- Nybrovej (4): The last one located on the roof of a house near across the lake.

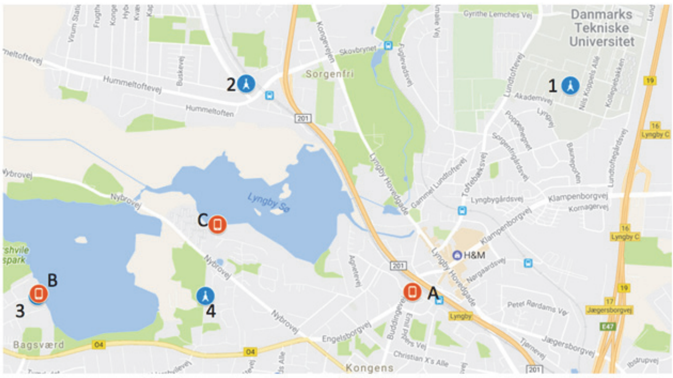


Fig. 6. Map of the gateways and spots

Fig. 6 also shows the static locations, which were selected taking into account the number of gateways receiving packets from the end-node:

- Lyngby building (A): Located at the highest floor of a high building. 454 samples were recorded.
- Bagsværd (B): Location set in front of the lake, next to a gateway (3). 1728 samples were recorded.
- Lyngby lake (C): The most centered spot, in the center of the four gateways. 168 samples were recorded.

The two different algorithms were applied in the three spots separately.

A. Extraction of TDOA

Fig. 7 displays the difference between two received times from two gateways in a temporal scale (identifiers of the packets). The actual value of the TDOA is also plotted in a red line which was computed before using the true position of the end-node in the spot. Therefore, all the obtained samples should be near this red line. However, one point is far from the real value, so it is a potential outlier. A periodicity in the signal can be observed and this is due to the microseconds accuracy of the internal clock of the gateways. The GPS data is acquired 1 Pulse-Per-Second (PPS), so every second it gets a correction. During this second, the internal clock is responsible to timestamp the packets. However, due to the otherwise accurate picosecond precision timestamp is sampled by the microsecond clock this gives rise to the observed periodicity.

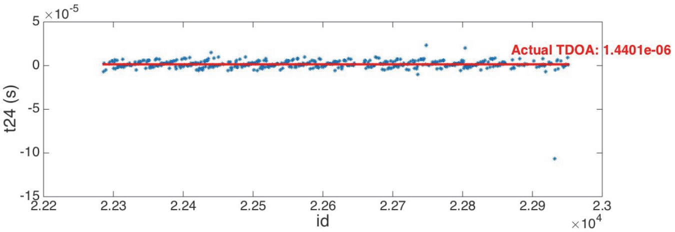


Fig. 7. TDOA extraction

B. *Detection of outliers*

The goal in this step was to remove the potential outliers in order to improve accuracy. The value that restricts how far the outliers are is the α parameter, called significance level, since it has influence in setting the upper bound. In other words, it is the probability of incorrectly rejected outliers. The optimum value of α was computed for the three different locations. The outliers were removed from the dataset. Fig. 8 shows the improvement made with that procedure. The samples that were far away from the actual TDOA disappeared.

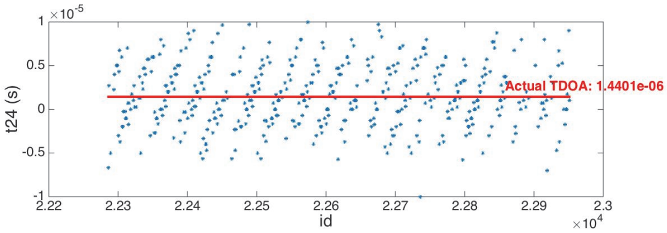


Fig. 8. TDOA extraction without outliers

C. *Non-iterative algorithm*

In order to obtain useful results from the estimated coordinates, the Mean Absolute Localization Error (MALE) was computed. The error consists in comparing the distance (error) between two coordinates using the Haversine formula, one versus one, and then doing the average of the whole distance error dataset. The coordinates were compared with the ones acquired from the GPS receiver in the three different spots.

TABLE I presents the results of the MALE in all the scenarios. In the first row, the overall dataset is used, while in the second row the outliers are removed. The error is fairly lower without outliers, although in the Bagsværd spot it is almost the same.

TABLE I. NON-ITERATIVE ALGORITHM MALE

	Lyngby building (A)	Bagsværd (B)	Lyngby lake (C)
MALE	5,54 km	10,72 km	9,76 km
MALE - No outliers TDOA	5,46 km	10,71 km	9,67 km

The values in table I were not accurate enough, so the mean estimator -also called sample mean- was computed to improve the results:

$$\bar{t}_{ij} = \frac{1}{N}\sum_{k=1}^{N} t_{ijk} \qquad (2)$$

Where t_{ijk} is the value of the TDOA t_{ij} for the sample k, and N is the size of the t_{ij} dataset. In this way, the algorithm only used the mean estimator of each TDOA. The results were considerably better (TABLE II). In this case, the error is called Absolute Localization Error (ALE), since the result is not an array as before.

TABLE III. NON-ITERATIVE ALGORITHM ALE. SAMPLE MEAN.

	Lyngby building (A)	Bagsværd (B)	Lyngby lake (C)
ALE - Sample mean TDOA	167 m	89 m	427 m
ALE - No outliers, Sample mean TDOA	116 m	65 m	206 m

D. *Iterative algorithm*

The grid of latitude and longitude coordinates was created near the area delimited by the gateways, as can be seen in Fig. 9. The iterative algorithm tested all the possible values inside this region. The step between latitude and longitude values was configured to 0.0001 degrees, which means around 11 meters of precision at the equator, increasing towards the poles.

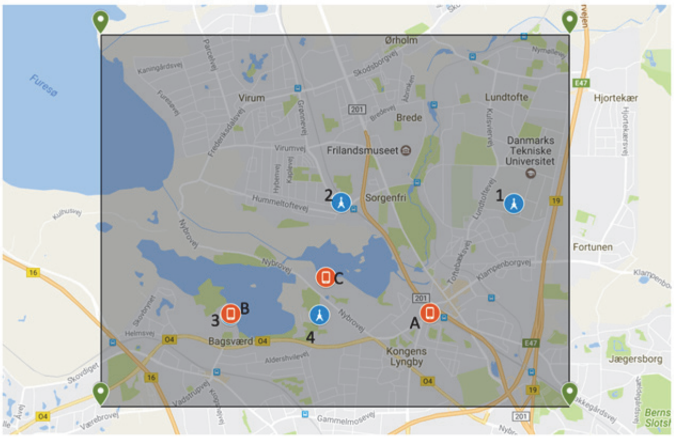


Fig. 9. Grid are of the iterative algorithm

The MALE (TABLE III) was computed in the same manner as in the previous algorithm. The results were better because the possible values of latitude and longitude were always inside the grid.

TABLE IV. ITERATIVE ALGORITHM MALE

	Lyngby building (A)	Bagsværd (B)	Lyngby lake (C)
MALE	1,39 km	1,09 km	1,15 km
MALE - No outliers TDOA	1,38 km	1,08 km	1,12 km

TABLE II. ITERATIVE ALGORITHM ALE. SAMPLE MEAN.

	Lyngby building (A)	Bagsværd (B)	Lyngby lake (C)
ALE - Sample mean TDOA	167 m	175 m	136 m
ALE - No outliers, Sample mean TDOA	119 m	127 m	114 m

The sample mean improved the accuracy of the results, as it is shown in TABLE IV. Removing outliers had a big impact when averaging, since the difference was larger.

The computational resources needed for the iterative algorithm are higher, so it should be difficult to use it in “real-time". However, as the algorithm is performed in the server site, the resources can be considered as unlimited, since nowadays great features are available.

VI. Conclusions

In this paper, a whole IoT tracking system was designed and implemented in order to present accuracy results using LoRa technology. The two designed algorithms demonstrated that it can be feasible to locate a device in a static spot with an accuracy of around 100 meters. However, for a real-time tracking application it can only be seen as a first approach, and not as a usable one. In general, the results of the iterative algorithm were better because the potential solutions of geodetic coordinates were restricted to a specific area. Nevertheless, the computational resources needed in such algorithm are much higher since all the coordinates are tested. But, as the algorithm is in the server site, this should not be a problem because the available resources there are unlimited. The Generalized ESD test allowed to detect the main outliers to improve the accuracy. The mean estimator clearly enhanced the results and both algorithms presented approximately the same values.

VII. Future Work

Some improvements must be implemented in order to make this device work as a real-time tracking device. The enhancements should not only be done in the algorithm, but also in the gateways, since they are responsible for recording the time when the packet is received. The clock of the Kerlink gateways was not intended for using in geolocation techniques so a more efficient and faster radio frontend would be beneficial by allowing to obtain more accurate timestamps. Another solution could be to increase the number of gateways in order to increase the number of TDOA and consequently improve the accuracy. Another problem is the multipath of the signal, which causes wrong measurements of the TDOAs. Gateways do not always receive the direct path of the signal due to the reflections with terrestrial objects like buildings, forests or mountains. The ability to resolve this phenomenon depends on the bandwidth of the signal. If the bandwidth is large, the resolution is better and vice versa. The bandwidth employed in LoRa is small (125 kHz), so the recorded times in the gateways can be the time of a multipath signal instead of the direct one.

Regarding the algorithm, other lines could be followed to improve the results. First of all, Machine Learning techniques like Decision Tree, Naive Bayes or Support Vector Machine could be applied to combine TDOAs with RSSI measurements. Low values of RSSI might mean that the received signal is not the direct path and can be discarded. In this case, a large dataset would be needed in order to divide it in three different groups: one for training, one for validating and one for assessing the suggested model. In order to improve the accuracy in "real-time", the k-Nearest Neighbors technique could be used, since it processes closer samples. The Haversine formula used in the non-iterative algorithm can be substituted by the Vicenty formula which is more accurate.

Acknowledgment

The authors would like to thank Indesmatech for borrowing the gateways and giving technical support during the whole project.

References

[1] L. Mainetti, L. Patrono, A. Secco, and I. Sergi, "An IoT-aware AAL system for elderly people," *2016 Int. Multidiscip. Conf. Comput. Energy Sci. Split. 2016*, 2016.

[2] © Libelium Comunicaciones Distribuidas S.L., "Waspmote LoRaWAN Networking Guide," p. 56, 2016.

[3] LinkLabs, "Low Power, Wide Area Networks," p. 16, 2016.

[4] "LoRa geolocation - Multipath." [Online]. Available: https://www.link-labs.com/lora-localization/.

[5] S. Sas, "Location-Enabled LoRa TM IoT Network : ' Geo -LoRa- ting ' your assets."

[6] A. Augustin, J. Yi, T. Clausen, and W. Townsley, "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," *Sensors*, vol. 16, no. 9, p. 1466, 2016.

[7] "Semtech." [Online]. Available: http://www.semtech.com/.

[8] A. Note, "Gateway to Server Interface LoRaWAN Network Server Demonstration : Gateway to Server Interface Definition Gateway to Server Interface," no. March, pp. 1–17, 2015.

[9] "Packet forwarder." [Online]. Available: https://github.com/Lora-net/packet_forwarder.

[10] "The Things Network (TTN)." [Online]. Available: https://www.thethingsnetwork.org/.

[11] V. Barnett; T. Lewis, *Outliers in Statistical Data*, 3rd ed. Wiley Series in Probability and Mathematical Statistics, 1994.