CHALMERS
UNIVERSITY OF TECHNOLOGY

Indoor positioning in LoRaWAN networks

Evaluation of RSS positioning in LoRaWAN networks using commercially available hardware

Master's thesis in Communication Engineering

RASMUS HENRIKSSON

Department of Signals and System
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2016

Master's thesis 2016:45

Indoor positioning in LoRaWAN networks

Evaluation of RSS positioning in LoRaWAN networks using commercially available hardware

RASMUS HENRIKSSON

Department of Signals and Systems
Division of Communication systems
Chalmers University of Technology
Göteborg, Sweden 2016

Indoor positioning in LoRaWAN networks
Evaluation of RSS positioning in LoRaWAN networks using commercially available hardware
RASMUS HENRIKSSON

Examiner: Henk Wymeersch, Signals and Systems

Master's Thesis 2016:45
Department of Signals and Systems
Division of Communication Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Göteborg, Sweden 2016

Indoor positioning in LoRaWAN networks
Evaluation of RSS positioning in LoRaWAN networks using commercially available hardware
RASMUS HENRIKSSON
Department of Signals and Systems
Chalmers University of Technology

Abstract

As the Internet of Things expands, new networks are launched in order to provide long range links with low power usage. While the low power consumption is one of the key components of devices connected to these networks, low cost is another which allows for large scale deployments of devices. By providing a position estimation built into these networks both power and cost can be cut by not having to use any GNSS solution. This thesis investigates if the received signal strength from LoRaWAN networks can be used to determine the position of a connected end-device. The study is done by simulating a large scale network with real world measurements to study how the number of anchor nodes impact the positioning accuracy. Results show that the measurements vary too much to give a good position estimation.

Acknowledgements

This work is the final part in my master's program Communication Engineering at Chalmers University of Technology. The work have been conducted for Cybercom at their Gothenburg office.
I would like to thank Gabriel Ibanez and Henrik Lundqvist at Cybercom for trusting me with this task and their support along the way. I would also like to thank Henk Wymeersch for being my examiner and helping to point me in the right direction during the project.

Rasmus Henriksson, Göteborg, June 2016

Contents

Acronyms

3GPP	3G Partnership Program
ADR	Adaptive Data Rate
AoA	Angle of Arrival
BW	Bandwidth
dBm	Decibell milliwatt
DL	Downlink
FHSS	Frequency-Hopping Spread Spectrum
FPGA	Field-Programmable Gate Array
FSK	Frequency Shift Keying
GNSS	Global Navigation Satellite System
IID	Independent and Identically Distributed
IoT	Internet of Things
ISM	Industrial, scientific and medical (radio bands)
LBT	Listen Before Talk
LLS	Linearised Least Squares (method)
LoRa	Long Range (modulation technique)
LoRaWAN	Long Range Wide Area Network (network standard)
LOS	Line Of Sight
LPWAN	Low Power Wide Area Network
M2M	Machine to machine (communication)
MAC	Medium Access Control
ML	Maximum Likelihood
RMS	Root Mean Square
RSS	Received Signal Strength
SF	Spread Factor
SNR	Signal-to-Noise Ratio
ToA	Time of Arrival
UL	Uplink
WAN	Wide Area Network
WSN	Wireless Sensor Network

1

Introduction

In the year 2016 everything seem to revolve around the emergence of IoT, where anything from cars to bathroom scales is to be connected to the internet in order to provide additional services for the consumer. However, the driving force behind IoT is more likely to be industry applications evolving from M2M technologies. Speaking in broad terms, IoT is simply an evolution of M2M communications, where more emphasis is put on connecting large number of nodes and using an ethernet back-end to reroute data as needed. This is a crucial step towards building smart city applications and what is called the fourth industrial revolution, where experts anticipate that the borders between what is physical, digital and biological will become blurred in industries [1]. With such a large scale goal and a lack of existing infrastructure to power it, companies are fighting to launch their own solution for networks supporting the IoT. While many M2M applications relied on 2G for large scale deployment, or smaller local networks for indoor, the IoT revolution sees the emergence of networks with different requirements, such as low chip-cost and support for tens of thousands of nodes in a single cell. Right now there are a lot of upcoming network standards that promise to solve these and many other issues. Among the most discussed competitors are Sigfox, LoRaWAN and LTE-M. This thesis will focus on the network standard called LoRaWAN, within the subset of networks dealing with long-range and low-power applications.

1.1 Low-Power Wide-Area Networks

LPWAN is an acronym for Low-power Wide-Area Networks, which as the name suggests, focus on providing large cells for low-power end-devices, typically battery powered. The main driving force is to provide network access for M2M and IoT applications over large distances, but still maintaining a low power consumption. On top of this, many companies are competing to provide cheap and easy-to-install solutions, making for quite a challenging task.

As range, power and bandwidth are closely related factors in networks, changing one of these will inevitably have an impact on the other two. In the case of LPWANs, the bandwidth is sacrificed in order to ensure a low-power communication over long range. What the terms 'low' and 'long' refers to in this context is not clearly defined, but a typical LPWAN have a range of about 10 km, an expected battery life for end-devices of 5-10 years and a maximum bit rate around 1-50 kpbs.

1.1.1 LoRaWAN and Competitors

Currently there are several LPWAN standards developed by different organisations. Of these LoRaWAN, Sigfox, NB-IoT, LTE-M and Weightless are probably the ones most often heard of. While there are large differences in how the standards are built in term of modulation, bit rate etc, the main differences are probably in the different business models [2]. Notably LoRaWAN and Weightless are both developed as open standards in unlicensed spectrum, meaning that anyone can set up their own network if desired. This opens up for interesting dynamics in a business area dominated by mobile network operators. On the other end of the spectrum, Sigfox is a company acting as both developer and operator, who are aiming for worldwide network support.

The last two standards, NB-IoT and LTE-M, are both standardized by 3GPP and developed in the traditional way that has formed the latest cellular network standards. The aim for these techniques are to provide a standard for network operator to deploy in licensed spectrum on top of existing 4G network. Additionally the evolution of 5G aims to target IoT applications as well, but as it is still in an early development stage it will not be considered further at this point in time.

1.2 Indoor positioning in research

In IoT in general, and particularly in wireless sensor networks, a core issue is power consumption and cost of modules. By deploying an internal positioning technique in the network instead of using a GNSS based technique, both of these factors can be kept to a minimum. Moreover many of these networks are intended for indoor use, eliminating the use of GNSS at all because they lack the required Line-of-Sight (LOS).

In current research there have been quite a few attempts to solve the problem of indoor positioning with sufficient accuracy with varying results. While there are commercial systems such as the iBeacon and WiFiSLAM from Apple they are often to crude to be used in industry, and are mainly developed for use in advertisement. In research however, WiFi has been used with an accuracy of 1m when using a combination of ToA/AoA (Time of Arrival, Angle of Arrival) and multiple precoded messages. By spacing these messages over the sample time in the receiver, a good estimation of the position can be calculated with existing hardware [3]. A complementary standard is Bluetooth Low Energy (BLE), where the main algorithm used is fingerprinting with beacons as anchor nodes [4]. With fingerprinting a heat map of the signal strength is constructed for the area in which the positioning algorithm is to be used. While this yields good results, the map will degrade over time and needs to be updated continuously.

1.3 Outline of project

1.3.1 Purpose

The purpose of this thesis is two-fold. The first is to make use of the LoRaWAN network standard to set up a functioning network. This network will then be used to gather measurements in order to simulate how a large-scale deployment would provide indoor positioning estimates. With this simulation, different aspects on how a positioning algorithm using LoRaWAN would function will be evaluated.

1.3.2 Objective

The objective of the thesis can be summarised as the four goals:

- To learn and utilise the open network standard LoRaWAN
- To build a functioning LoRaWAN network to use for gathering of measurements
- Simulate a positioning algorithm as a proof-of-concept
- Evaluate the accuracy of the positioning system, and suggest improvements for further studies

1.3.3 Scope

The project is set to deliver a positioning method using existing LoRaWAN hardware. The method is intended as a proof-of-concept, and therefor aspects such as cost and battery consumption are disregarded.

1.4 Thesis outline

The thesis is divided into six chapters including this one. Chapter 2 will explain how the LoRaWAN standard functions, and elaborate on how the modulation called LoRa is a vital part of this network standard.
Chapter 3 will go through the basics of wireless position techniques. It will also relate these techniques to the LoRaWAN standard and evaluate which are of interest for the research question.
Chapter 4 describes the methods used for each step of the project. The hardware of the network is presented, how measurements were conducted and finally how the simulation was carried out.
Chapter 5 will present the results of both measurements and simulation, along with a discussion on interpretations. Improvements and fault sources will also be dealt with in this chapter.
Finally, chapter 6 will present a conclusion and suggestions for how further research can be conducted on this subject.

2
LoRaWAN

LoRaWAN is an LPWAN developed and maintained by the LoRa Alliance. This standard builds upon the LoRa modulation developed by Semtech and adds a network layer to handle traffic between end-devices and central nodes. This allows for communication over long range for low data-rate devices. The main motivation behind the network is to enable IoT, wide-area sensor networks, and other M2M applications. For the current release (v1.1), the focus of the network is on uplink communications.

2.1 Organisation

The LoRaWAN standard is developed by a non-profit organization called the LoRa Alliance. The LoRa Alliance include over 200 companies as of June 2016, which span the whole LoRaWAN ecosystem. The organisation is split around sponsors, contributors, institutional members and adopters, where the first three categories mainly work with development, and the last is certified to sell LoRa-compatible products. Initially the LoRa standard, short for Long Range, was developed by a French startup called Cycleo in 2012, which was later acquired by Semtech. Semtech also founded the LoRa Alliance together with companies such as Microchip, IBM and Cisco with the goal of creating a network standard using the LoRa modulation. Currently Semtech is the owner of two patents used in the LoRa chipset, *Fractional N-synthesized chirp generator* [5] and *Low power long range transmitter* [6], which makes Semtech a key player in the LoRa Alliance.

2.2 Modulation

The key enabling factor in LoRaWAN networks is the LoRa modulation standard. The LoRa modulation uses a proprietary Chirp Spread Spectrum (CSS) scheme, which creates wideband linear frequency modulated chirps. The chip rate of these chirps are equal to the spectral bandwidth of the signal and uses 125, 250 or 500 kHz of bandwidth. The gains of using using CSS are twofold, the first being that chirps are noise resistant and the second that these chirps can be generated with high precision using a cheap crystal, which leads to low chip costs. Because of the relative broadband characteristics of the chirps, multi-path fading is typically not an issue [7]. Doppler spread causes a frequency shift, which also only have a small effect on the channel thanks to the time-varying frequency of the chirps. When

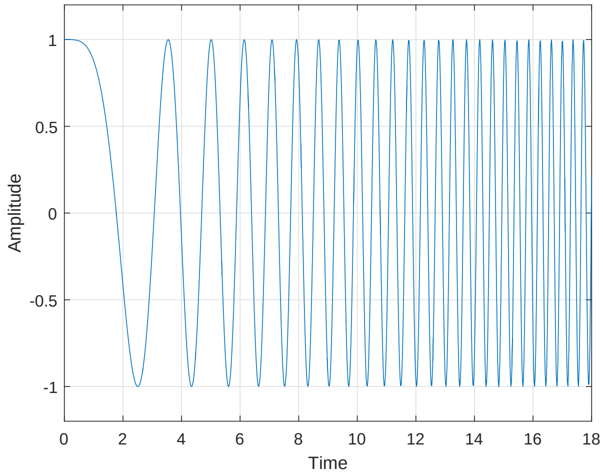

Figure 2.1: Visualisation of the up-chirps used in the LoRa modulation. The frequency increases as a linear function of time.

using the LoRa modulation, 15 km of range can be achieved in urban environments and up to 30 km with good line-of-sight [7].
Additionally LoRa uses a Frequency-Hopping Spread Spectrum (FHSS) scheme to switch frequency between available channels according to a pseudo-random distribution. This helps to further mitigate interference.
A key thing to note with the CSS modulation scheme is that it produces a very sharp peak when auto-correlated, and has previously been deployed in radar applications [8]. The high peak helps to identify the correct time that the signal is received, and thus can be used to give a good estimate of the time it takes for a transmission to travel between two nodes.

2.2.1 Symbol coding and bit rate

The bit rate of LoRa varies between 0.3 and 22 kpbs depending on the spread factor used. The spread factor can take on values between 7 and 12, and each of these are associated with a set of orthogonal codes which allows for simultaneous communication at different bit rates. Symbol coding in LoRa is accomplished by using time-shifted up-chirps which are interleaved to improve robustness [9]. By using a higher spreading factor, the robustness of the communication link is increased but as a consequence the time-on-air also increases.

2.3 Network structure

The LoRaWAN network protocol is a standard developed by the LoRa Alliance. The first version was released in 2015, and the current version as of June 2016 is 1.1. The network uses the terminology gateways, end-devices and network server to distinguish between different entities. The end-devices are typically some kind of sensor, which are further divided into three different categories depending on

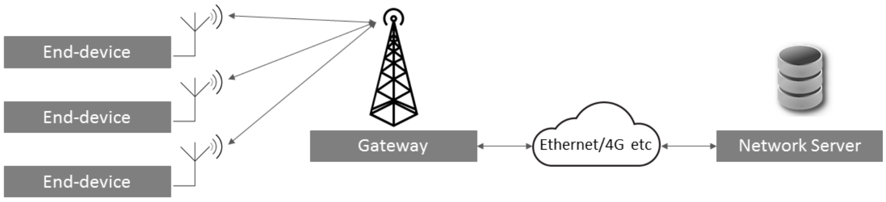

Figure 2.2: Sketch of a typical LoRaWAN network setup.

their power consumption. The end-devices are connected to gateways in a star topology, where the gateway is the central node. Typically a gateway can handle several thousands of end-devices, however they do not communicate directly with other gateways. Instead gateways are connected by ethernet to the network server, which stores data and handles traffic in both directions. This makes for a star-of-star topology where the network server is the central node, and gateways are intermediate nodes. As LoRaWAN operates in unlicensed ISM bands, gateways are typically set up by both companies and private users. Currently there are a few crowdsourced networks such as The Things Network. In these networks users share a network server provided by a company, and gateways are crowdfunded locally and open to use for anyone who own LoRa compatible end-devices.

2.3.1 Device classes

LoRaWAN devices are divided into three different classes depending on their intended load on the network. All devices must have at least Class A functionality to be considered LoRa certified devices. Restrictions for Class A devices are to open two receive windows following each uplink transmission. In practice this means that the device can be inactive for long periods of time to conserve battery power. The only way for the gateway to communicate downlink with the end-device is to wait for an uplink transmission and then respond.
Class B devices have, on top of Class A functionality, dedicated time slots for receiving downlink messages, and also periodically receives beacon messages for clock synchronisation. Class C devices are continuously listening, and are therefore intended to be connected to a power supply.

2.3.2 License-free carrier

The LoRa modulation is built around using a license-free carrier frequency which varies between different regions of the world. Currently there are standards for Europe, US and China which span certain ranges within 433-870 MHz. All of the used carrier frequencies in LoRaWAN are ISM bands. For Europe, 863-870 MHz is the most common band used, which will also be used in the studies in this project. Because LoRaWAN is not based on a Listen Before Talk (LBT) principle, there are certain restrictions limiting the duty cycles between 0.1-10% depending on which band is used.

3

Positioning

Positioning is crucial for many LPWAN applications due to the nature of the data gathered. Specially as the number of devices can range up to several thousands within a single cell, estimating position is of vital importance in order to install and use data from end-devices properly. An example of this is using temperature sensors to measure temperature fluctuations in an urban area. As the number of temperature sensors increase, so will the accuracy of the city's temperature distribution model. However, when using a large number of sensors it will become tedious and expensive work to manually program the position of each sensor.

A natural solution to this problem would be to equip each sensor with a GNSS tracker, for example using GPS. While this solution is tempting, adding a GPS tracker to a device will increase both cost and power consumption [10]. Another issue with using a satellite based system is the lack of indoor coverage, which does not only include offices and home, but also factories and malls. Taking this into consideration, speculation can be made that the LPWAN to be used in the future is the one which first can introduce a reliable positioning system which uses low power and works indoor.

3.1 Positioning basics

In this chapter the terminology anchor nodes will be used to describe nodes with an a priori known location and target nodes are nodes with unknown position which will have their position estimated. To simplify calculations, all nodes are considered to exist in a 2-dimensional plane and thus in a real-world scenario this corresponds to all nodes being located at the same altitude.

To find the position of target nodes one must first identify which measurable parameters are of interest, and then link these parameters mathematically to the distance between anchor and target nodes. Typically these are the received signal strength (RSS), ToA or Aoa, which can be utilised in a variety of different ways to calculate the distance between nodes. As AoA requires antenna arrays to function, which is not supported in the LoRaWAN specification as of now, it will not be considered further.

When RSS or ToA are used, the most simple approach to visualise the system of equations is that shown in equation (3.1), where p_i is the position of anchor nodes, p_t the position of the single target node, and v_i is the measurement error associated with the distance between p_i and p_t. Note that v_i is a random variable, which is assumed to be zero-mean [11]. The real properties of v_i can be obtained by executing test runs where p_t is known.

$$|p_i - \hat{p}_t| = \sqrt{(x_i - \hat{x}_t)^2 + (y_i - \hat{y}_t)^2} + v_i \tag{3.1}$$

$$\begin{cases} p_i = (x_i, y_i), \quad i = 1, .., n_a \\ \hat{p}_t = (\hat{x}_t, \hat{y}_t) \end{cases} \tag{3.2}$$

To solve equation (3.1) the task is to maximise the probability density function $p(p_t|\hat{p}_t)$ where $\hat{p}_t$ is the estimated position given by measurements. This can either be done in non-iterative ways such as linearised least squares method, or by using an iterative filtering function suitable for non-linear equations such as an extended or linearised Kalman filter. For the sake of estimating the positioning accuracy, the more simple method of least squares estimation will be used in this project.

3.2 Received Signal Strength

As wireless signals traverse air they are subject to several effects causing signal degradation. Usually these are described as a sum of a second-order function and a random variable which describes time and frequency related fluctuations. In the simplest case scenario, the power received P_r is related to the power transmitted, P_t, through the free space path loss model described in equation (3.3). The FSPL model consider only LOS connections, and model these as deterministic processes.

$$P_r = P_t \left(\frac{\sqrt{G_r G_t} \lambda}{4 \pi d} \right)^2 \tag{3.3}$$

Which rewritten in decibel scale becomes the following equation:

$$P_r[\text{dBm}] = P_t[\text{dBm}] - 20 \log_{10}(d) - 20 \log_{10}(\frac{4\pi}{\lambda}) \tag{3.4}$$

Through the free space path loss model it is clear that the power received depends not only on the distance between nodes, but also the antenna gain, G, and the wavelength of the carrier signal, λ. Through empirical studies the FPSL model have evolved into what is known as the log-distance path loss model, where the unknown parameters are determined through experiments [11]. The parameters are determined at a reference distance, d_0, and clump together into a single parameter, PL_0 as seen in equation (3.5). Please note that all power units will be in the dBm scale from now onwards.

$$P_r = P_t - PL_0 - 10\gamma \log_{10} \left(\frac{d}{d_0} \right) \tag{3.5}$$

Here γ denotes the path loss exponent, and varies depending on environment and hardware setup. In the previous FSPL model this exponent is equal to 2.
The last piece in the RSS equation is the shadowing effect, denoted X_g, which is a random variable to describe fluctuations in measurements caused by various disturbances such as interference from other transmissions, weather effects or scattering.

The distribution of the shadowing effect is often modeled as a zero-mean Gaussian random variable in logarithmic scale, which corresponds to a Rayleigh distribution in linear scale. This gives the final expression of the received signal strength, which once it has been characterised properly can be used to calculate the distance between two nodes. In the end, determining the position comes down to a trilateration problem where three anchor nodes are needed to find a unique solution in a 2-dimensional space.

$$P_r(d) = P_t - PL_0 - 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + X_g \tag{3.6}$$

3.3 Time of Arrival

Time of Arrival, also known as Time of Flight (TOF), is the process of determining distance from the time a transmission takes from anchor node to target node. In theory this is a straight forward procedure since the speed of light is well known. The distance between two nodes are calculated from the time difference between transmitting and receiving, and as a result the position can be determined by trilateration in the same way as in the RSS case. However in practice, this becomes a lot harder due to clock drifts. Essentially the problem comes down to clock synchronization, where nodes need to be synchronized down to nanosecond scale in order to achieve a proper distance approximation. For a network such as LoRaWAN where the nodes are supposed to be low-cost and idle for a large amount of the time, the internal clock drift makes this a quite hard problem.

There are techniques to go about this, such as two-way time of arrival (TW-TOA) or time difference of arrival (TDOA). The common denominator of these techniques are that they only demand the anchor nodes to be time synchronised, and therefor the low cost of target nodes is not compromised. In TW-TOA the round trip time between anchor and target node is measured, and if the target node have a well defined processing time of the message this can give a good distance estimation. For TDOA the target node sends a broadcast message which is received by multiple anchor nodes. The anchor nodes, which are time synchronised, can then calculate the distance from the difference in time between signal receptions. This is a multilateration problem which involves solving a set of hyperbolic functions, and therefore an additional anchor node is needed compared to the trilateration case.

3.4 Calculating position estimates

The act of calculating position estimates from given measurements and models is without doubt a whole area in itself, with its' own inherent difficulties. This thesis will focus on using a linearised least squares method to provide a rough idea on what magnitude the positioning error achieves, and relate this to the number of anchor nodes available.

3.4.1 Non-iterative algorithm: Linearised Least-Squares

Let d_i be the distance between anchor node i and the target node, and $\hat{d}_i$ the estimated distance. The relationship between these two variables can be described by equation (3.7) where v_i is an unknown random variable corresponding to the positioning error. By writing out the vectors d_t and d_i as their $x-$ and y-coordinates it is clear that this is not a linear equation and will need some type of linearisation in order to be used in a least-square sense. By evaluating the squares introduced in the previous step, equation (3.9) is found.

$$\hat{d}_i = d_i + v_i \tag{3.7}$$

$$(\hat{d}_i - v_i)^2 = (x_i - x_t)^2 + (y_i - y_t)^2 \tag{3.8}$$

$$(x_t^2 + y_t^2) - 2(x_i x_t + y_i y_t) = (\hat{d}_i - v_i)^2 - (x_i^2 + y_i^2) \tag{3.9}$$

This equation is rewritten by introducing the variables R_i and R_t as the nodes' radii from origo. This gives equation (3.11).

$$\begin{cases} R_t = \sqrt{x_t^2 + y_t^2} \\ R_i = \sqrt{x_i^2 + y_i^2} \end{cases}$$

$$\Rightarrow R_t^2 - 2(x_i x_t + y_i y_t) = \hat{d}_i^2 - 2\hat{d}_i v_i + v_i^2 - R_i^2 \tag{3.10}$$

$$\hat{d}_i^2 - R_i^2 = -2x_i x_t - 2y_i y_t + R_t^2 + 2d_i n_i - n_i^2 \tag{3.11}$$

The final step is now to rewrite this as a linear system, which is done with vector notation in equation (3.12). The j-index denotes row j of the matrix.

$$\mathbf{h} = \mathbf{G}\theta + \mathbf{n} \tag{3.12}$$

$$\begin{aligned} \text{where} \quad \theta &= [x,\ y,\ R^2]^T \\ \mathbf{G}_j &= [-2x_i,\ -2y_i,\ 1] \\ \mathbf{h}_j &= [\hat{d}_i^2 - R_i^2] \\ \mathbf{n}_j &= [v_i^2 - 2\hat{d}_i v_i] \end{aligned} \tag{3.13}$$

Since this equation is linear, the least-square estimator is known as

$$\hat{\theta} = \left(\mathbf{G}^T \mathbf{W} \mathbf{G}\right)^{-1} \mathbf{G}^T \mathbf{W} \mathbf{h} \tag{3.14}$$

The price to pay for linearisation is that the measurement noise v is squared to produce n. Assuming that v is a zero-mean variable this should still produce a reliable estimation.

4

Implementation

In this chapter methods used to build the network, conduct measurements and simulate position estimation will be described. Also any hardware used in the network will be described. For simulations, Matlab have been used.

4.1 Network setup

For this project one gateway and one end-device have been used to gather measurements of signal strength. The gateway is named Lorank-8, and manufactured by the company Ideetron. The gateway consist of a Beaglebone Black controlling an iC880A concentrator module from IMST. The Beaglebone board acts as a host for the concentrator, which forwards any LoRa-based traffic onto a wired ethernet connect, and vice versa in the case of downlink messages. The iC880A board uses a Semtech SX1301 baseband processor to modulate the symbol stream, and is capable of handling up to eight connections simultaneously [12]. The Lorank-8 is shipped with the appropriate software and can be considered plug-and-play for the purpose of this thesis.

For the purpose of end-nodes, an Arduino based solution was built. The setup consists of an Arduino Uno, paired with a LoRaWAN module from Libelium, connected through a multi-protocol shield from the same company. The LoRaWAN module is based on the RN2483 board from Microchip, and is shipped with appropriate firmware and API library. This end-device have been programmed as a Class A device for the sole purpose of sending uplink messages, and the source code is presented in Appendix A.

The network back-end consist of a server hosted by Semtech for testing purpose, where both transmission content and metadata can be downloaded. For any device, parameters such as RSS, SNR, SF and BW of the signal can be extracted.

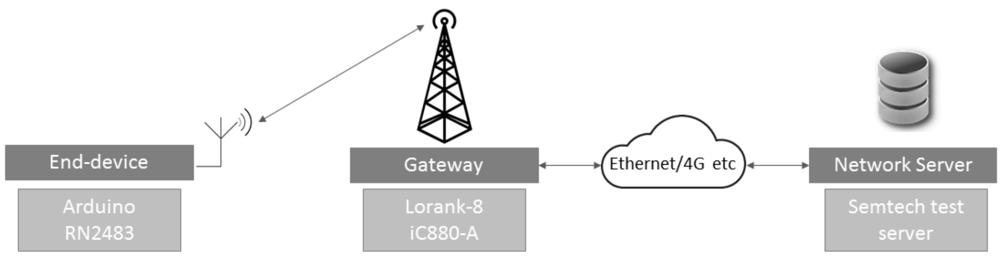

Figure 4.1: Network map of implemented LoRaWAN setup.

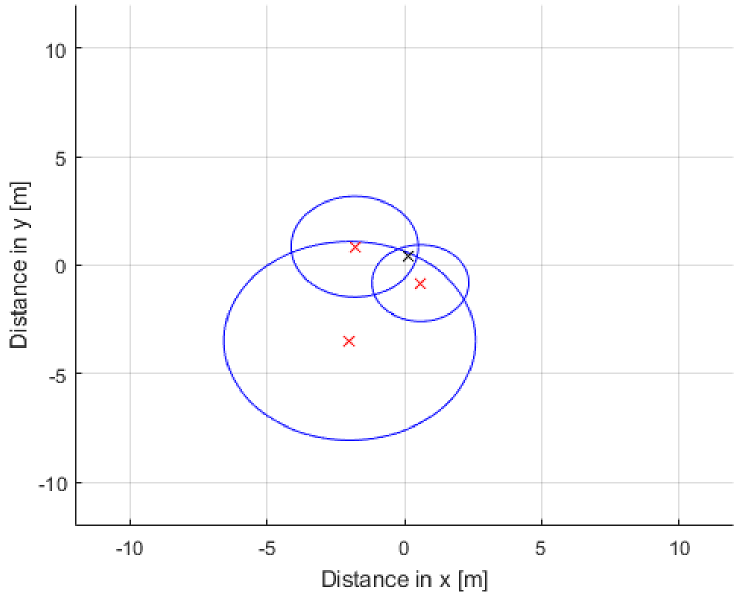

Figure 4.2: Overview of the simulation setup. The red crosses are anchor nodes, which have blue circles representing the calculated distance between anchor and target node. The black cross is the estimated position using LLS, and its' distance from origo shows the position error.

4.2 Gathering RSS measurements

To gather indoor measurements, both gateway and end-device are placed within an office area. The office area used for testing have a typical open layout with a cubicle type design. By alternating the placement of the end-device, several measurement series are gathered at different distances. All series are made for LOS condition, and at certain distances multiple series are gathered to try and validate the log-distance path loss model in eq. (3.6). The measurement data is then used to find the model parameters, which in the simulation will be used to determine the distance between gateway and end-device. The RSS measurements are presented as integers in dBm scale.

4.3 Simulation

To evaluate the accuracy of this positioning approach Matlab have been used to conduct simulations where the number of anchor nodes vary. The goal of the simulations is to put a number on how good the accuracy is for this method of positioning. To get a rough understanding of this, a non-iterative optimisation method is used to calculate mean and standard deviation as a function of nodes. The simulation uses four general steps which are described below.

1. In this setup, the real position of the end-device is always at $(0, 0)$. n gateways are generated at random distance and angle from origo, both drawn from a uniform distribution. While the angle can have any value in the range $[0, 2\pi]$, the distance may only take on discrete values which have measurement data associated with them. For this simulation, it is assumed that all antennas are isotropic radiators.
2. Each gateway, $g_i, i \in [1, n]$, is associated with an RSS value depending on the real distance between the end-device in origo and g_i. The RSS value is drawn from the set of actual data associated with this distance, and the index of this value is from a uniform distribution.
3. The theoretical distances, $\hat{d}_i$ are calculated by using eq. (3.6) with the extracted model parameters. This model assumes that the noise is zero-mean, and as such the true distance is associated with a random variable as in eq. (4.1).

   $$d_i = \hat{d}_i + v \tag{4.1}$$

   From each gateway, located at (x_i, y_i), a circle is drawn with radii $\hat{d}_i$. A sketch of how the simulation setup looks can be found in figure 4.2.
4. The problem of finding the true position is now a non-linear optimisation problem, where the point closest to all circles is to be found. This is solved by using the linearised least squares method presented in equation (3.12).

5

Results and discussion

This chapter will present the results found from the measurements and simulations carried out during this project, followed by a discussion interpreting them. First the measurement data will be presented and analysed. These measurements are then used in the simulations presented in previous chapter. The results of these simulations are presented with focus on how the positioning accuracy can be increased with a growing number of anchor nodes. The results of these simulations are then discussed in the second part of this chapter.

5.1 Indoor measurements

The indoor measurements are gathered using the network setup described in chapter 4. The first plot, figure 5.1, shows how two different measurement series gathered at the same distance differ in their histogram. The two series are gathered at a distance of 3m from the gateway, but with different angles. Both series are in LOS, and they are spaced in time by one day. As seen in the plot, there is a substantial difference in the mean power received, which varies by 4.4 dB, representing an average energy increase of 2.75 times. The whole indoor measurement range is made up of samples

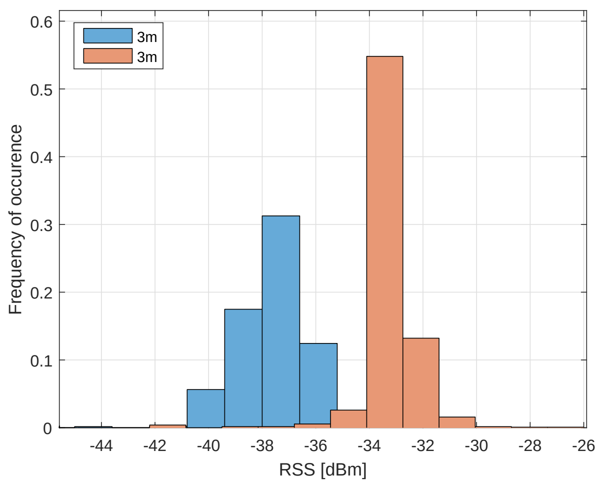

Figure 5.1: Histogram of received measurement strength at a distance of 3 meters. The measurement series are spaced in time by 24 hours. As seen in the figure, the measurements seem to vary either with time or position.

between 1 and 11 meters, taken at different positions within the office area. Due

to the hardware cap on messages sent per time unit in LoRa, together with how the gateway handles lost transmissions, the number of measurements in each series varies. The number of individual measurements per series is found in figure 5.3.
In figure 5.2 the mean power received is plotted together with error bars to denote the standard deviation of the measurements. Note that since $d = 6$m have quite few measurement points compared to the other series, the true standard deviation might differ. While the last three points in figure 5.2 does not seem to agree with

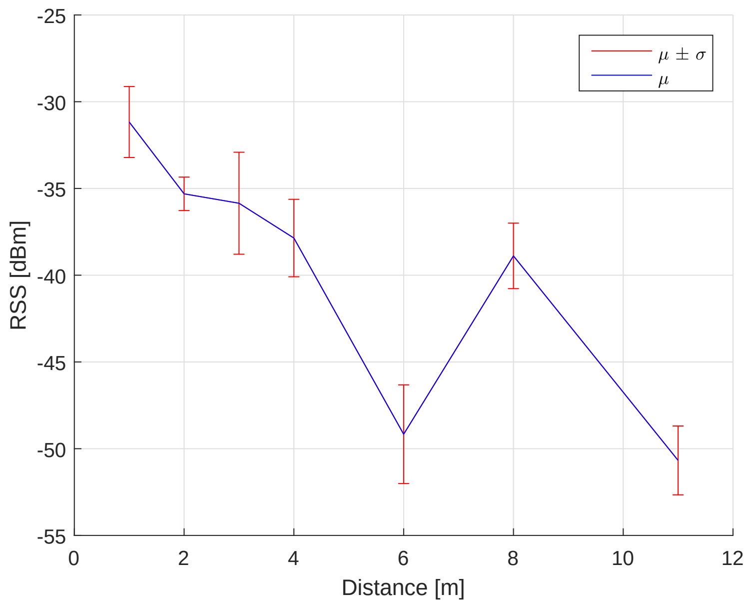

Figure 5.2: Mean power received from the end-node as a function of distance between gateway and end-node. Error bars show the standard deviation for each mean value.

the model presented earlier, a key feature to note is that the standard deviation does not increase with the distance.

Figure 5.3: Table showing the total number of measurements per distance, as well as the number of individual measurement series are used.

d [m]	nr of samples	nr of series
1	6248	3
2	978	2
3	2274	2
4	1117	1
6	332	1
8	971	1
11	1131	1

To extract the factors γ and PL_0 from equation (3.6), the distance $d_0 = 3$m is used. This gives the least square estimation shown in figure 5.4, where the measurement mean for each d is plotted with a logarithmic distance of base ten. Here, $\gamma = 1.69$ and $PL_0 = -35.9$.

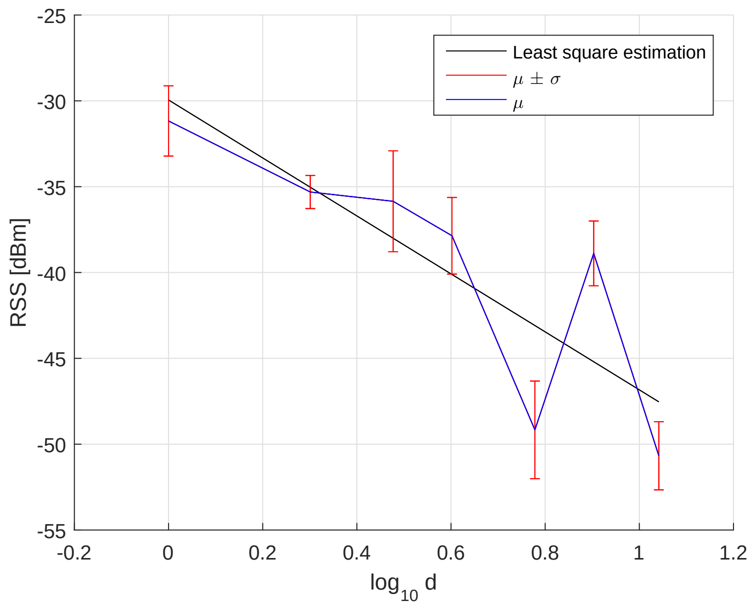

Figure 5.4: RSS plotted against a logarithmic distance. The black graph is the least square estimator of the measurements when correlated with the log distance path loss model.

By studying the deviation of measurement values from the distance-related mean value, $\mu(d)$, a distribution for the shadowing effects can be found. This is done by subtracting $\mu(d)$ from each measure point, and results in the histogram in figure 5.5. From visual inspection, it is clear that the histogram bear strong resemblance to a Gaussian distribution, however the additional peaks suggest that it is the sum of several Gaussian random variables.

5.2 Simulation results

The positioning simulation are carried out as a function of the number of nodes, n. At each $i \in [3, n]$ the root mean square (RMS) error and its' standard deviation is calculated over 1000 iterations. The result of this simulation is shown in figure 5.6. From the figure it can be concluded that the accuracy of the position estimation increase with the number of nodes used. In figure 5.7 the RMS error and standard deviation are plotted for up to $n = 100$. An interesting fact when studying these figures are that the RMS error fall below 10 meters at $n = 40$, and the measurements are conducted only in a range up to 11 meters. Also at 40 nodes, the standard deviation is still around 8 meters.

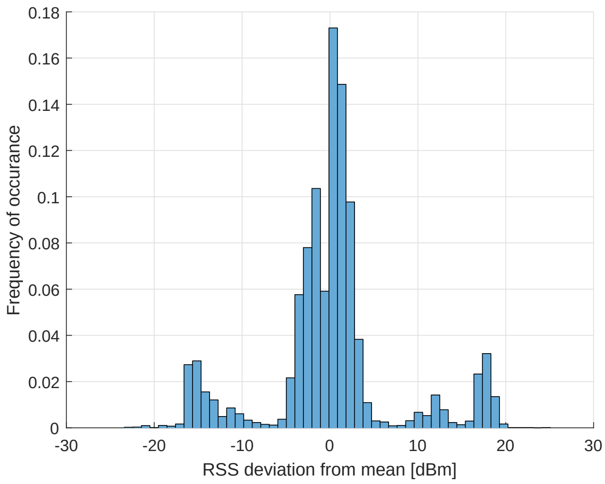

Figure 5.5: Histogram over the shadowing effect for all measure points. Mean value is equal to zero, and standard deviation is equal to 7.36.

5.3 Measurement and simulation interpretations

When studying the simulation results there is a clear tendency that both the mean and variance of the RMS error decreases as the number of nodes increase. This is not a surprising result given the zero mean characteristic of the shadowing effect, however the rate of decline is not sufficient to build a functional positioning system. Judging from the results in figure 5.7, the RMS error seem to have a steady decline even at $n = 100$. This, in theory, suggests that the positioning algorithm will improve as the number of nodes inside the network grows. However, the problem in using LoRaWAN is the star topology, which does not allows communication between end-devices. In a real world scenario, having access to more than 10 gateways at one time is uncommon. This would cap the error at 20 ± 22 meters of accuracy, which does not give any additional information about the position of the end-device considering the measurement range. However, if communication between end-devices was to be allowed, this approach might work in very large networks. This positioning would probably have the trade-off of using more power to locate all nodes, but for some scenarios it might be beneficial. For large scale networks however, the distances will probably be greater than 11 meters, and thus to be sure that this approach works it might be a good idea to perform a study over a larger distance.
Studying the simulation results presented in figure 5.1 it is clear that the signal strength varies not only with distance, but also with position. This is most likely due to interference or multipath effects, and have the consequence that one measurement series have significantly higher values. Using the mean values of these two measurement series in the path loss model suggested in equation (3.6), results in a distance of either 2 or 3.5 meters. As the true distance is 3 meters, neither of these values are correct and will thus give a faulty distance estimation to the position algorithm, even though they are taken as mean values over several hundred measurements. This result is not surprising, and shows that fingerprinting might be a technique to evaluate when building indoor positioning with LoRa.

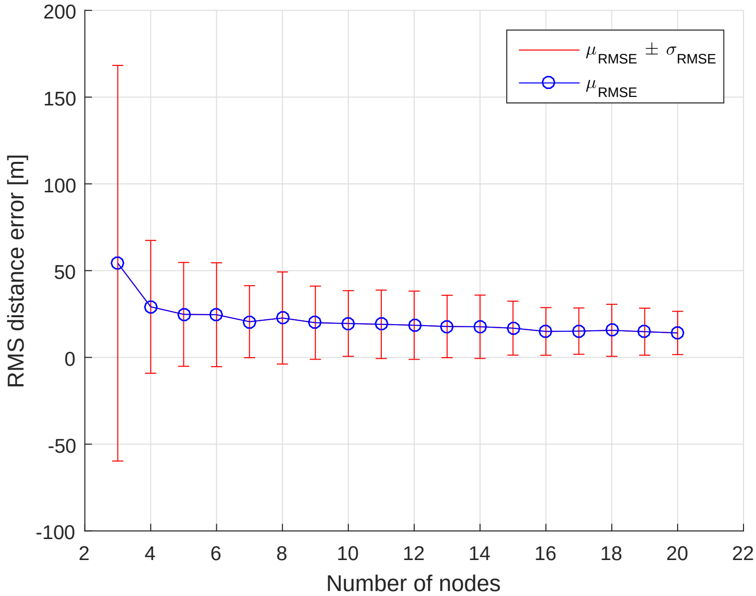

Figure 5.6: Simulated RMS error of position estimation with an increasing number of anchor nodes. For each point on the x-axis 1000 iterations are used to calculate the mean and standard deviation. From the graph it is clear that the position error decreases as the number of anchor nodes increase.

5.4 Time of arrival or received signal strength

The advantage of using RSS instead of TOA is the reduced complexity of the hardware needed. In these scenarios a time drift of 10 ns correspond to a drift in distance of 3 meters. The radio board of the gateway used in this project have a maximum sampling frequency of 36 MSamples/s [13], corresponding to a sampling time of 28 ns. This would, with a perfect clock synchronisation, give a ranging error of ± 4.2 meters, which is on a slightly lower magnitude than the error from using RSS.

In the scenario of this project there are two main issues with implementing a ToA solution. The first is that the clock of the SX1257 LoRaWAN board is dependent on a crystal oscillator which is very likely to drift over time. The alternative is using the Arduino board to supply a clock signal, which is not reliable down to nanosecond scale. The second problem is the black box characteristics of purchasing a commercial LoRaWAN board. The board is supplied with a set of APIs to relay instructions, and these are not built with the intention of supplying a high accuracy time reference for the signals received.

However there are a lot of benefits with using TOA together with a LoRa modulation, which should be evaluated in future research. The technique have good synergies with the chirp modulation, which is both resistant to multipath effects and have good auto-correlation properties for determining the timestamp of the received signals. This has been proven earlier in [14] [15] [16], but not explicitly for a LoRa modulation.

If a LoRa system using ToA would be constructed, the most viable solution is

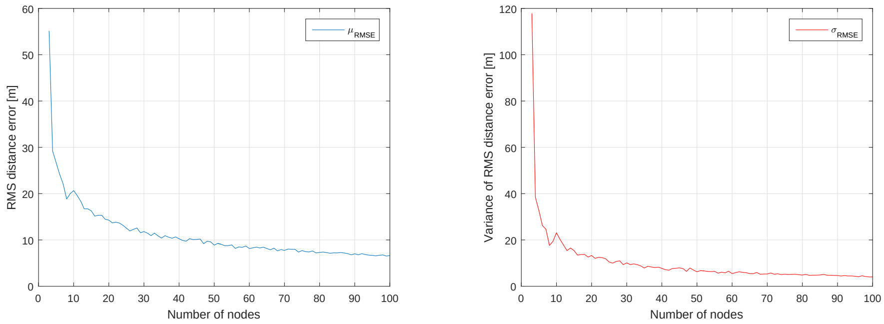

Figure 5.7: Mean and standard deviation of the RMS error as a function of number of nodes. The values are calculated from a series of 100 measurements at each number of nodes.

probably to use TDOA. By time synchronising the gateways and utilising a TDOA scheme, the low complexity of the end-devices can be maintained in favor for a low price tag.

5.5 Positioning algorithm

The positioning algorithm used in this project is a rather crude estimation compared to other algorithms available. While there are many more non-iterative algorithms that solve the ML estimation with better results, the one used in this project is mainly chosen to give a hint on what the accuracy would be in a large scale deployment. There is also the option to use iterative algorithms to solve this problem such as Kalman or particle filters. These are more computational heavy, but also known to give good results even with non-linear problems.

6

Conclusion

To conclude this report, using RSS measurements to create a positioning estimation does not give satisfactory results in an indoor environment. Even when simulating 100 anchor nodes, the mean square error is 8 meters. When the conducted measurements are in the range of 1-11 meters, this does not give much information about the real position of the target node. In a real world scenario this can probably give information in which general direction the target node can be found, but for any application requiring a real position estimate another approach should be considered. That being said, the approach used here might be of use in larger outdoor scenarios. However due to time constraints, it has not been tested in this study.
For future studies it would be wise to focus on the more reliable ToA approach for a number of reasons. The largest reason is of course the increased accuracy, but also with ToA the asymmetrical cost sharing between gateway and end-devices can be upheld. With the help of a few gateways acting as anchor nodes with high time accuracy, a type of TW-ToA or TDOA scheme can be used to position several thousand of nodes with only small changes to the LoRaWAN scheme.
Other approaches include using mesh networks for the large number of measurements, or utilising AoA, of which none are supported in the current LoRaWAN standard. The most pressing issue however is probably using a more complex positioning algorithm. In any real world scenario the the LLS method used in this project is far too crude and it is possible that a Kalman or particle filter with an appropriate motion model can give far better position estimates using the same set of data.

Bibliography

[1] K. Schwab, "The fourth industrial revolution: what it means, how to respond," *Foreign Affairs*, 2016.

[2] N. Hunn, "LoRa vs LTE-M vs Sigfox," 2015.

[3] C. Yang and H.-R. Shao, "WiFi-based indoor positioning," *IEEE Communications Magazine*, no. March 2015, pp. 150–157, 2015.

[4] R. Faragher and R. Harle, "An analysis of the accuracy of Bluetooth low energy for indoor positioning applications," 8-12 September 2014 2014.

[5] C. A. Hornbuckle, "Fractional N-synthesized chirp generator," 2010.

[6] O. B. A. Seller and N. Sornin, "Low power long range transmitter," 2013.

[7] J. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hänninen, and M. Pettissalo, "On the coverate of LPWANs: Range evaluation and channel attenuation model for LoRa technology," 2015.

[8] Semtech, "An1200.22 LoRa modulation basics," May 2015 2015.

[9] B. Sikken, "Decoding lora," 2016.

[10] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location," *IEEE Signal Processing Magazine*, no. July 2005, pp. 24–40, 2005.

[11] A. Goldsmith, *Wireless Communications*. Cambridge: Cambridge University Press, 2005.

[12] GmbH.

[13] Semtech, "Sx1257 rf front-end transceiver datasheet," 2012.

[14] P. Ferrari, A. Flammini, E. Sisinni, A. Depari, M. Rizzi, R. Exel, and T. Sauter, "Timestamping and ranging performance for IEEE 802.15.4 css systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 5, pp. 1244–1253, 2014.

[15] Y. Zhang, "Precise location technology based on chirp spread spectrum," *Journal of Networks*, vol. 6, no. 6, pp. 872–878, 2011.

[16] L. Huang, Y. Lu, and W. Liu, "Using chirp signal for accurate RFID positioning," 28-30 July 2010 2010.

[17] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios." 2015.

[18] J. Cheng, Y. Cai, Q. Zhang, J. Cheng, and C. Yan, "A new three-dimensional indoor positioning mechanism based on wireless lan," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[19] H. Cho and S. W. Kim, "Mobile robot localization using biased chirp-spread-spectrum ranging," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 8, pp. 2726–2835, 2010.

[20] Ericsson, "Cellular networks for massive IOT (white paper)," January 2016 2016.
[21] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios," *IEEE Signal Processing Magazine*, no. July 2005, pp. 70–84, 2005.
[22] M. R. Gholami, *Positioning Algorithms for Wireless Sensor Networks.* Thesis, 2011.
[23] M. R. Gholami, S. Gezici, E. G. Ström, and M. Rydström, "Hybrid TW-TOA/TDOA positioning algorithms for cooperative wireless networks," 5-9 June 2011 2011.
[24] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks," *IEEE Signal Processing Magazine*, no. July 2005, pp. 41–53, 2005.
[25] K. J. Krizman, T. E. Biedka, and T. S. Rappaport, "Wireless position location: Fundamentals, implementation strategies, and sources of error," May 5-7, 1997 1997.
[26] D. Macagnano, G. Destino, and G. Abreu, "Indoor positioning: a key enabling technology for iot applications," 2014.
[27] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localizatio tehchniques," *Computer Networks*, vol. 51, no. 10, p. 2529–2553, 2007.
[28] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes," *IEEE Signal Processing Magazine*, no. July 2005, 2005.
[29] P. Pivato, S. Dalpez, and D. Macii, "Performance evauation of chirp spread spectrum ranging for indoor embedded navigation systems," *7th IEEE Interation Symposium on Industrial Embedded Systems*, pp. 307–310, 2012.
[30] S. Sand, A. Dammann, and C. Mensing, *Positioning in Wireless Communcation Systems.* John Wiley and Sons, 2014.
[31] Semtech, "An1200.13: LoRa designer's guide," July 2013 2013.
[32] Semtech, "LoRa FAQs," 2015.
[33] Semtech, "Sx1272/73 datasheet, rev. 3," 2015.
[34] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent, "LoRaWAN specification v1.0," January 2015 2015.
[35] G. Sun, J. Chen, W. Guo, and K. R. Liu, "Signal processing techniques in network-aided positioning," *IEEE Signal Processing Magazine*, no. July 2005, pp. 12–23, 2005.
[36] K. Yu, I. Sharp, and Y. J. Guo, *Ground-Based Wireless Positioning.* Wiley-IEEE Press, 2009.

A

End-device source code

For the latest version, please visit https://github.com/rashen/lora

```
/*       LoRaWAN Network Test v0.3
*        Rasmus Henriksson
*        rasmhenriksson@gmail.com
*/

// Cooking API
#include <arduinoUtils.h>
#include <arduinoUART.h>
#include <arduinoMultiprotocol.h>
#include <arduinoClasses.h>

// LoRaWAN
#include <arduinoLoRaWAN.h>

// Bluetooth serial
//#include <SoftwareSerial.h>
//SoftwareSerial btSerial(9, 10); // RX,TX
//String command = "";
//String message;

// Pins
const int errorLed = 13;
//const int btVcc = 8;

// Constants
uint8_t socket = SOCKET0;
uint8_t port = 1; // Range 1-223
int packetCount = 0;

// Device parameters for Back-End registration
char DEVICE_EUI[] = "0000000000000000"; // Insert device
   EUI
char DEVICE_ADDR[] = "00000000"; // Insert device address
char NWK_SESSION_KEY[] = "2
   B7E151628AED2A6ABF7158809CF4F3C"; // Configured for
```

```
   TTN
char APP_SESSION_KEY[] = "2
   B7E151628AED2A6ABF7158809CF4F3C"; // Configured for
   TTN
char APP_KEY[] = "00000000000000000000000000000000"; //
   Insert unique (and secret!!) key here

// Variables
uint8_t error;
uint8_t SNR;
char data[] = "00010203040506070809"; // Payload to send
uint32_t cFreq[] = { 867100000, 867300000, 867500000,
   867700000, 867900000 };
uint32_t fOffset = 200000;
int errorCount = 0;

void showError(uint8_t e, int n) {
        if (e == 0) {
                digitalWrite(errorLed, LOW);
        }
        else {
                //btSerial.println("Error occured");
                for (int i = 1; i <= n; i++) {
                        digitalWrite(errorLed, HIGH);
                        delay(100);
                        digitalWrite(errorLed, LOW);
                        delay(100);
                }
        }
}

void softwareReset() {
        asm volatile ("␣␣jmp␣0");
        //wdt_enable(WDTO_15MS);
}


void setup() {
        pinMode(errorLed, OUTPUT);
        //pinMode(btVcc, OUTPUT);
        //btSerial.begin(9600);

        //digitalWrite(btVcc, HIGH);
        digitalWrite(errorLed, LOW);

        // 1. Activate LoRaWAN
```

```
        error = 1;
        LoRaWAN.ON(socket);
        LoRaWAN.factoryReset();

        // Channel parameters
        for (uint8_t i = 3; i < 8; i++) {
                LoRaWAN.setChannelStatus(i, "on");
                LoRaWAN.setChannelFreq(i, cFreq[i - 3]);
        }
        for (uint8_t i = 0; i < 3; i++) {
                LoRaWAN.setChannelDutyCycle(i, 302);
        }

        for (uint8_t i = 3; i < 8; i++) {
                LoRaWAN.setChannelDutyCycle(i, 99);
        }

        //btSerial.println("Setting channel parameters");
        LoRaWAN.setPower(5); // [N/A, 14, 11, 8, 5, 2]
           dBm
        LoRaWAN.setADR("off"); // Adaptive data rate
        LoRaWAN.setDataRate(5); // [250, 440, 980, 1760,
           3125, 5470, 11000];



        // Set device EUI and address
        error = LoRaWAN.setDeviceEUI();
        LoRaWAN.setDeviceAddr(DEVICE_ADDR);
        showError(error, 2);


        // Keys and retries
        LoRaWAN.setNwkSessionKey(NWK_SESSION_KEY);
        LoRaWAN.setAppSessionKey(APP_SESSION_KEY);
        LoRaWAN.setAppKey(APP_KEY);
        LoRaWAN.setRetries(3);

        // 8. Save config
        error = LoRaWAN.saveConfig();
        showError(error, 5);

        LoRaWAN.joinABP();

}
```

```
void loop() {
        //LoRaWAN.joinABP();
        error = LoRaWAN.sendUnconfirmed(port, data);
        showError(error, 2);
        if (error != 0) {
                errorCount = errorCount + 1;
                if (errorCount > 3) {
                        LoRaWAN.joinABP();
                        if (errorCount > 50)
                                softwareReset();
                }
        }
        else
                errorCount = 0;
        delay(3000);
        port = errorCount + 1;
}
```