Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jernej Grosar

Meritev pokritja in prodora signala nizkoenergijskega prostranega omrežja

magistrsko delo

magistrski program druge stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Fabio Ricciato

Ljubljana, 2017

University of Ljubljana
Faculty of Computer and Information Science

Jernej Grosar

Coverage and penetration measurements for Low Power Wide Area Network signals

master's thesis

the 2nd cycle master's study programme
computer and information science

Supervisor: Assoc. Prof. Fabio Ricciato

Ljubljana, 2017

Declaration of master's thesis authorship

I, the undersigned Jernej Grosar am the author of the Master's Thesis entitled:

Coverage and penetration measurements for Low Power Wide Area Network signals

With my signature, I declare that:

- the submitted Thesis is my own (unaided) work which was done under the supervision of Assoc. Prof. Fabio Ricciato,
- all electronic forms of the Master's Thesis, the title (Slovenian, English), the abstract (Slovenian, English) and the keywords (Slovenian, English) are identical to the printed form of the Master's Thesis,
- I agree with the publication of the electronic form of the Master's Thesis in the collection "Dela FRI".

in Ljubljana, 11th September, 2017 Author's signature:

Acknowledgments

I would like to express my sincere gratitude to my mentor Assoc. Prof. Fabio Ricciato for guiding me through this thesis. I would also like to thank Nikola Jovalekic for developing LoRa module for Raspberry Pi 3 used in this thesis and to Assist. Prof. Veljko Pejović for all the help that he provided to me. I would also like to offer a word of thanks to my girlfriend Kaja, who provided me with unfailing support and helped me run the measurements. Last but not the least, I would like to thank to my family which has faith in me and has been a source of considerable support all these years.

Jernej Grosar, 2017

Contents

List of Acronyms

AES	Advanced Encryption Standard
AP	Access point
API	Application Programming Interface
BW	Bandwidth
CR	Coding rate
CRC	Cyclic redundancy check
CSS	Chirp spread spectrum
DIO	Digital input output
FSK	Frequency-shift keying
GPRS	General Packet Radio Service
GPS	Global positioning system
IOT	Internet of things
ISM	Industrial, scientific and medical
JSON	JavaScript object notation
LoRa	Long Range Radio
LoRaWAN	LoRa wide area network
LTE-M	Long Term Evolution - M
M2M	Machine-to-machine
MAC	Media access control
MQTT	Message Queue Telemetry Transport
OSI	Open Systems Interconnection
RPMA	Random Phase Multiple Access
SDR	Software defined radio
SF	Spreading factor
SIM	Subscriber identification module
SPI	Serial Peripheral Interface
SQL	Structured Query Language

Povzetek

Brezžična senzorska omrežja (ang. wireless sensor networks, krajše WSN) so iz dneva v dan bolj razširjena in igrajo vedno večjo vlogo v našem življenju. Danes ta omrežja večinoma uporabljajo več skočne tehnologije kratkega dometa kot so ZigBee, Z-Wave ali Bluetooth, vendar so te nepraktične, če hočemo z njimi zagotoviti pokritost večjega območja ali celotnega mesta. Pred kratkim se je, kot rešitev tega problema, pojavila nova tehnologija imenovana nizkoenergijsko prostrano omrežje (ang. Low Power Wide Area Network, ali krajše LPWAN). Ta ima, v primerjavi z brezžičnimi senzorskimi omrežji, boljši domet, daljšo življenjsko dobo baterije (naprava lahko delujejo več let z eno samo baterijo) in večinoma uporablja eno skočno brezžično povezavo. Ena takih tehnologije je tudi radio dolgega dometa (ang. Long Range Radio, krajše LoRa), okoli katere je zgrajena ta magistrska naloga.

Cilj našega dela je bil razviti fleksibilno, prilagodljivo in odprtokodno platformo za kratkoročno ali dolgoročno merjenje pokritja in prodora signala nizkoenergijskega prostranega omrežja. Po končanem razvoju raziskovalne platforme smo se osredotočili na merjenje pokritosti in prodora LoRa signala v različnih okoljih. Pri vseh teh meritvah smo, zaradi lažje kasnejše primerjave rezultatov med različnimi okolji, uporabili enake kombinacije nastavitev LoRa modulacijskih parametrov. Večinoma smo se osredotočili na meritve v zaprtih prostorih in na ulicah, naredili pa smo tudi nekaj testov dometa LoRa signala. Najboljši rezultat (najdaljši domet signala), ki smo ga dosegli pri teh meritvah, je bil 39 kilometrov. Na tej točki je bil signal še vedno močen, sprejem paketov pa prav tako zadovoljiv, vendar smo morali zaradi

tehničnih težav meritev prekiniti.

Ključne besede

LoRa, nizkoenergijsko prostrano omrežje, internet stvari, odprtokodno, raziskovalna platforma

Abstract

Wireless sensor networks (WSN) are evermore widespread and play from day to day a bigger role in our lives. Up to date, such networks use multi-hop short-range wireless technologies such as ZigBee, Z-Wave or Bluetooth. These technologies are impractical to use, if one wants the coverage of an area in the size of a city. Lately, new technologies called Low Power Wide Area Networks (LPWAN) emerged. They have greater range, longer battery life (they can work on a single battery for a few years) and they mostly use single-hop wireless communication. One such technology, which we will focus on in this master's thesis, is a Long Range Radio or LoRa for short.

The goal of our work is to develop an open source, free to use, scalable research platform for conducting short or long term LoRa signal penetration and coverage measurements. With this platform finished, we focused on performing a LoRa signal penetration and coverage measurements in different environments. For all these measurements the same sets of LoRa modulation parameter settings were used, so one can easily compare the signal propagation in different environments. We mainly focused on urban and indoor measurements, but a few range tests were also conducted. In one of these range tests, the distance of 39 kilometers was achieved. At this point, the signal was still quite strong and the packet reception was still satisfying, but due to technical problems, the measurement had to be finished.

Keywords

LoRa, Low Power Wide Area Networks, Internet of Things, open source, research platform

Razširjen povzetek

Brezžična senzorska omrežja igrajo v našem življenju iz dneva v dan vedno večjo vlogo. Prisotna so že vsepovsod: v industriji, medicini, prometu in tudi v kmetijstvu. Večina današnjih brezžičnih senzorskih omrežji ima kratek domet (na primer ZigBee ali Bluetooth), zato so nepraktična, če hočemo z njimi zagotoviti pokritost večjega območja ali celotnega mesta.

Danes je vse več naprav povezanih v tako imenovani internet stvari (ang. Internet of Things). Za povezovanje teh naprav se večinoma uporablja GPRS, vendar ta počasi izginja, 3G in 4G modemi pa so precej dražji. Pred kratkim se je na trgu brezžičnih senzorskih omrežji pojavila nova tehnologija imenovana nizkoenergijsko prostrano omrežje (ang. Low Power Wide Area Network, ali krajše LPWAN), katera obljublja, da bo odpravila problem pokritja večjih površin. Te naprave imajo veliko večji domet, tudi do nekaj deset kilometrov, kar pomeni, da z le peščico njih pokrijemo celotno mesto, poleg tega pa so tudi energetsko varčne, saj lahko delujejo več let z eno samo baterijo. Te tehnologije za svoje delovanje večinoma uporabljajo omrežno topologijo imenovano zvezda-zvezd (star-of-star). Ena takih tehnologij je tudi radio dolgega dometa (ang. Long Range Radio ali krajše LoRa), katero smo uporabili v naši magistrski nalogi.

Najbolj znane in razširjene tehnologije, ki sodijo v skupino LPWAN omrežij so: LoRaWAN, SigFox, LinkLab, Ingenu in Weightless. Vsaka ima nekaj prednosti in slabosti, katero bo kdo uporabil, pa je odvisno od njegovih potreb. Pri izbiri le te mora upoštevati domet, podatkovno prepustnost, življenjsko dobo baterije, koliko naprav je lahko največ priklopljenih v omrežje

in seveda ceno.

Kot smo že omenili, smo pri izdelavi naše magistrske naloge uporabili LoRa tehnologijo. Ta spada v fizično plast OSI referenčnega modela in za svoje delovanje uporablja tehniko žvižgajočega širjenja spektera (Chirp spread spectrum oz. CSS), katera je odporna na Dopplerjev efekt in omogoča uporabo cenenih oscilatorjev. Pri uporabi te tehnike se podatki modulirajo s pomočjo širokopasovne linearne frekvenčne modulacije, pri kateri se frekvenca skozi čas zvišuje ali znižuje glede na modulirane podatke. Posebna lastnost LoRe je ta, da je posamezen bit kodiran z več "žvižgi"(ang. chirps), kar omogoča še posebno odpornost signala na motnje. S spreminjanjem pasovne širine (ang. bandwidth), faktorja razširitve (ang. spreading factor) in stopnje kodiranja (ang. coding rate) lahko nadzorujemo pretočnost omrežja in domet signala. Ti dve lastnosti sta obratno sorazmerni: če zvišujemo pretočnost omrežja, se znižuje domet signala in obratno.

LoRa prostrano omrežje (LoRaWAN) je podskupina LoRa modulacije, ki uporablja le del LoRa modulacijskih parametrov in je definirana na mrežni plasti OSI referenčnega modela. Naprave povezane v to omrežje delimo v tri razrede: razred A, B in C. Razred A vsebuje naprave, ki imajo po končanem oddajanju odprti še dve naključni kratki časovni okni za sprejem podatkov, razred B ima odprto, poleg dveh naključnih kratkih časovnih oken, še eno okno ob vnaprej določenem času in razred C, katerega naprave po oddajanju ves čas sprejemajo pakete.

V okviru pričujoče magistrske naloge smo izdelali fleksibilno, prilagodljivo in odprtokodno platformo za merjenje pokritja in prodora signala nizkoenergijskega prostranega omrežja, ki je dostopna na GitHubu [42]. Za izdelavo le te smo uporabili Raspberry Pi 3 Model B naprave, katere se uporabljajo kot sprejemniki in oddajniki (senzorji), Android aplikaciji (prva za beleženje pozicije premikajočega se senzorja v zaprtem prostoru in druga za beleženje GPS pozicije senzorja na prostem), ter strežnik, ki je namenjen shranjevanju podatkov in razpošiljanju ukazov preko MQTT protokola do naprav, ki so povezane na našo raziskovalno platformo.

Kot smo že omenili, smo v okviru te magistrske naloge uporabili naprave Raspberry Pi 3 Model B, tako za končne naprave (end-devices), kot tudi bazne postaje (gateways). Na vsako od teh naprav je pritrjen LoRa modul s Semtech SX1276 LoRa čipom, ki omogoča oddajanje in sprejemanje LoRa paketov. Na teh napravah se izvaja Python skripta, katera sprejema ukaze za delovanje naprave (poslane preko MQTT protokola) in izvaja podproces v katerem teče program (napisan v programskem jeziku C), ki pošilja in sprejema LoRa pakete. Statusi teh paketov se shranjujejo v medpomnilniku dokler ni ta poln, nato pa jih skripta, zapakirane v JSON sporočilu, s pomočjo MQTT protokola pošlje na strežnik. Ta sporočila se na strežniku razčlenijo in shranijo v SQL podatkovno bazo. Statusi služijo kot pomoč pri ugotavljanju razlogov zakaj paketi niso bili sprejeti in so sledeči: rxOnGoing (čakanje na pakete), signalDetected (paket je bil zaznan), headerInfoValid (ta status pomeni, da je glava paketa veljavna - ima veljaven CRC), OK (paket sprejet ali poslan) in CRCERR (paket je poškodovan - CRC ni veljaven).

Posebnost naše raziskovalne platforme je tudi, da lahko končne naprave in bazne postaje delujejo v enem izmed treh načinov: samo kot oddajnik (transmitter), samo kot sprejemnik (receiver) ali pa kot oboje "hkrati"(bi-directional mode). V zadnjem načinu naprava preklaplja med sprejemnikom in oddajnikom v vnaprej določenih časovnih obdobjih. Ta način je bil uveden, za namene testiranja dvosmerne povezave med oddajnikom in sprejemnikom. Vse nastavitve, za vse naprave priključene na to razvojno platformo, je mogoče shraniti v eno JSON datoteko in jo, z enim samim ukazom v terminalnem oknu, poslati na vse naprave. Poleg končnih naprav in baznih postaj, sta bili razviti tudi dve Android aplikaciji. Prva omogoča shranjevanje GPS koordinat poti po kateri se je premikala končna naprava na prostem, druga pa omogoča shranjevanje "poti"v zaprtih prostorih. Ta aplikacija v resnici beleži le čas, ko naprava doseže vnaprej določeno točko na poti. Z uporabo več takih strateško razporejenih točk v stavbi lahko nato lažje narišemo zemljevid pokritosti s signalom.

Ključnega pomena je tudi strežnik, na katerem tečejo Python skripte za

razčlenjevanje JSON sporočil poslanih iz končnih naprav, baznih postaj in Android aplikacij in shranjevanje le teh v SQL podatkovno bazo. Na tem strežniku deluje tudi MQTT strežnik, kateri skrbi za pošiljanje ukazov na naprave.

Ko je bil razvoj raziskovalne platforme skoraj končan, smo izvedli dva testa dometa signala. Prvi test smo izvedi v Ljubljani, kjer smo postavili oddajnik na okensko polico v šestem nadstropju in se s sprejemnikom peljali v smeri proti Brezovici. V tem testu smo dosegli razdaljo 4.1 kilometra, končna točka pa ni bila v vidnem polju (line of sight) oddajnika. Na tej razdalji je bil signal zelo šibek, prejetih pa je bilo le nekaj paketov. Po nekaj popravki kode smo izvedli drugi test. Tokrat smo oddajnik postavili na okensko polico planinske koče pod vrhom Matajurja, na nadmorski višini 1320 metrov in se s sprejemnikom nameščenim na strehi avtomobila odpravili v Italijo proti mestu Palmanova. V tem primeru je bil signal celotno pot zelo močen, večino poti pa je bilo v vidnem polju (line of sight) oddajnika. Pri tem merjenju smo dosegli razdaljo 39 kilometrov, katera bi lahko bila še daljša, če ne bi morali, zaradi tehničnih težav, prekiniti meritev. Na tej razdalji je bil signal še zelo močen in sprejem paketov odličen, dokler je bil sprejemnik v vidnem polju oddajnika, če pa je bila vmes kakšna ovira (na primer hiša), sprejema paketov skorajda ni bilo.

Po koncu razvoja raziskovalne platforme smo naredili še tri vrste meritev. Pri prvi meritvi smo raziskovali kako vplivajo različne postavitve oddajnika na odboj signala (in pokritost z njim) ob in za sosednjo stavbo. Tudi v tem primeru je bil oddajnik postavljen v šestem nadstropju, najprej na eni strani (dlje od sprejemnika za dvema armiranima stenama), nato pa bližje sprejemniku. Sprejemnik je bil v obeh primerih postavljen na strateških točkah okrog sosednje stavbe na višini enega metra. Drugi tip meritev smo izvedli na hodnikih Fakultete za računalništvo in informatiko v Ljubljani, pri čemer smo merili kako močen je signal na hodnikih v različnih nadstropjih stavbe. Pri teh meritvah smo testirali tudi dvosmerno komunikacijo, pri čemer sta bila sprejemnik/oddajnik in kontrolni sprejemnik v tretjem nadstropju na

južnem delu hodnika, z drugim oddajnikom/sprejemnikom pa smo se sprehajali po hodnikih v vseh treh nadstropjih, od juga proti severu. Moč signala smo primerjali za dve skupini nastavitev LoRa parametrov. Tretjo vrsto meritev smo izvedli na ulicah Ljubljane, pri čemer smo oddajnik/sprejemnik in kontrolni sprejemnik postavili na vrh Rožnika in se z drugo napravo zapeljali po ulicah Ljubljane. Prav tako, kot v prejšnjem testu, je bil tudi tu uporabljen dvosmerni način komunikacije in dve različni skupini nastavitev LoRa modulacijskih parametrov.

Vse te meritve so pokazale, da je postavitev oddajnika zelo pomembna za pokritost signala, prav tako so pomembne tudi nastavitve LoRa parametrov, kateri zelo vplivajo na pokritost in domet signala. Pri testiranju dvosmerne povezave so meritve pokazale, da je bil signal v smeri od premikajoče se naprave proti stacionarni napravi nekoliko boljši, prav tako je bilo več prejetih paketov. Kaj je razlog za to, da je sprejem paketov v eni smeri boljši, kot pa v drugi ne vemo, kajti morali bi opraviti še dodatne meritve. Sumimo, da je glavni razlog, da je prišlo do takšnih razlik ta, da se napravi (stacionarni in premikajoči sprejemnik/oddajnik) razlikujeta med seboj. Obe napravi sta namreč imeli pri vseh meritvah enako vlogo (ista naprava je bila pri vseh meritvah stacionarni sprejemnik/oddajnik).

Chapter 1

Introduction

Wireless sensor networks (WSN) play from day to day a bigger role in everyday life. We use them almost everywere: in industry, medicine, transportation and even in agriculture. Up till now, the main focus of WSNs has been to on develop technologies that are capable of sending data with high bit rate while still using as little power as possible. Most such technologies uses multi-hop short-range wireless technologies such as ZigBee and Bluetooth [1]. They provide solutions for low battery consumption, but are unpractical to use if one wants to achieve the coverage of an area the size of the a city. Using 2G/GSM, UMTS or LTE is usually not a solution either because of high hardware price or and signal transmission costs.

The new emerging and promising solutions are Low Power Wireless Sensor Networks (LPWAN). These devices are capable of sending data a few tens of kilometers away with very low transmission rate, so with only a few of them one can achieve the coverage of a whole city. Due to low bit rate such devices are capable of working on single battery for a few years and they are perfect for monitoring remote locations [2].

There are s few different LPWAN technologies on the market today [1], the one we will concentrate on in this thesis is Long Range Radio technology or LoRa [2] for short. It is very recent, has (partially) open standard and works in the unlicensed sub-GHz frequency bands (434 MHz or 868 MHz in

Europe and 929 MHz in the USA).

Today, there are some few papers published with evaluation of LoRa signal range and penetration capabilities, but none of them have delivered any research platform or conducted measurements with the same LoRa parameter settings in different environments. With that in mind, we have developed open source, free to use, scalable research platform for testing LoRa range and penetration capabilities.

This research platform consists of end-devices and gateways (which are in our case Raspberry Pis 3 Model B), Android applications for logging moving end-devices position and central control server for saving data and controlling measurements. All end-devices, gateways and Android applications are connected to the central control server via IP connection, while for the communication and distribution of commands, MQTT communication protocol is used.

Within this research platform, we also planned to use software defined radio (SDR) for more extensive analysis of LoRa signal and for better understanding why packets were not detected or decoded. Today, a few LoRa SDR implementations exists [3], [4] and [5], but despite authors claim that these implementations work, we were unable to send or decode any LoRa message.

The best results were achieved using [4], where packets were detected, but not correctly decoded. Despite that, we wanted to know either this software could be modified in the way that we could monitor packet status (if packet was detected, if LoRa header was correct and if it was decoded) or why did its reception fail. This was the time we tried to use SDR (we do not have any knowledge in this field), thus we did not fully understand algorithms used for packet detection and decoding so we could not perform modifications needed. Because of that, we decided to use Semtechs SX1276 chips modem statuses instead (see section 5.1.3 for details), which were a great alternative.

When research platform was nearly developed, two preliminary range tests were done, one in Ljubljana and one in Italy. After few modifications and extensive testing, final measurements were conducted. This in-

cludes "grid" measurements, where we measured signal propagation behind the building and indoor and urban coverage measurements. For all these final measurements, the same LoRa parameter settings were used for better understanding how a different environment effects signal propagation. All results are presented in chapter 6.

Chapter 2

Related work

LPWAN technologies are relatively new. When we started with this master's thesis (a year ago), there had been only a few measurements of their performance and range done, but up till now, quite a few of them have emerged.

Indoor penetration tests of LoRaWAN were performed, using commercially available devices strapped to researchers arms [6]. They measured data for 10 days while they were following their usual daily routine. Another indoor testing was been done in [7], using Raspberry Pi 2 and IMST IC880A LoRaWAN Concentrator. They measured signal reception and its strength on different fixed locations through the building. In Padova, Italy, a real life deployment of in building sensors using LoRa network connectivity is being used [8], where LoRa base station is placed on the 9th floor of the a 19 story building. It provides excellent coverage for the whole building, but unfortunately authors did not state which settings are being used. In paper [9], authors measured differences in building coverage when receiver is located on the roof and when it is located in the basement of a concrete reinforced multistory building. They found out that better coverage is achieved when receiver is placed on the roof, rather than in the basement.

In paper [10], authors tested how the strength and quality of the received signal is changing in urban and sub-urban area with respect to the antenna elevation, distance from the receiver, different antennas on mobile nodes and

different spreading factors used for packet transmission. Outdoor test, using fixed antennas and three different settings of a spreading factor, is presented in [11], but the data was measured only in a few places in the city. A much more extensive measurements of urban coverage and reliability tests have been done on the campus of Glasgow Caledonian University, using different spreading factors [12].

Both *Dong Hyun Kim at al.* [13] and *Wirnol San-Um at al.* [14] implemented a system for tracking objects using GPS and LoRa communication channel. While authors of the first article implemented bicycle tracking system, authors of the second one studied possibilities of using LoRa technology (with 433 MHz antenna) for tracking tactical troops. In this article, architecture for such an application has also been proposed.

As seen from [15], LoRa technology is also suitable for monitoring sailing boats in a bay.

Papers [16], [17], [18] and [19] present analysis of maximum number of LoRa end-devices connected to one base station when using different LoRaWAN parameters. Authors of the paper [17] have also tested LoRa packet reception under Doppler frequency shift (with the different angular and linear velocity), the coverage of a base station and mobile end-device range. The last two testes, the coverage and the range test, are presented in paper [20] by the same authors. A mathematical model of interference effects form multiple packets when using same parameters settings, on a single LoRa gateway is presented in [21].

Algorithm for power and parameter settings optimization was presented in [22]. It is capable of optimizing transmission power and LoRa parameters based on packets RSSI. These optimizations are aiming for the minimum battery usage of the system as a whole.

A proof of how little energy LoRa devices use, can be seen in [23], where authors present the vibration powered system for monitoring bridges. This system can gather enough energy from bridge vibrations in just 3.5 hours, to send one LoRa packet.

In paper [24], there is a description of how a flexible and scalable LoRa network server based on OpenStack can be built. This server is able to send and receive LoRa messages, store them into database and analyze received data, but it is not able to control end-devices.

All the above papers conducted measurements only in one or two environments i.e., in urban and sub urban area or in urban area and on the see. They did not measure how the signal propagates in many different environments while using the same parameters settings, or how the signal strengthens and packet reception rate changes, if the transmitter and the receiver change their roles (if bi-directional communication is used). Moreover, none of them delivered any open and scalable researching platform for long term measurements.

Chapter 3

LPWAN technologies and Internet of Things platforms

As the Internet of things market is evermore evolving, there is also greater battle in the field of infrastructures that supports it. Today, most of IoT devices connected to the network use GPRS modems and SIM cards. As GPRS networks are disappearing and because 3G or 4G modems are more expensive, new technologies like LoRaWAN, SigFox, LinkLab, Ingenu and Weightless are evolving. All these new technologies are so called Low Power Wide Area Networks or LPWAN for short.

Which technology one will chose depends mainly on their needs. The most important things that have to be taken in to account when choosing between these technologies are range (coverage), throughput, battery life, scalability and, of course, price.

For this master's thesis work LoRaWAN (LoRa in particular) was chosen because it has a great range, cheap hardware and is not dependent on any software. We can also explore a great range of different parameter settings and a custom research platform can be built.

3.1 LPWAN

LPWAN provides network connectivity for devices that do not require high bandwidth rate eg. sensors or security devices. LPWAN has a greater range, can operate at lower costs, has lower deployment costs and extended support for massive number of devices, unlike other consumer technologies such as Bluetooth, ZigBee, Z-Wave or WiFi [32].

Due to the short range of consumers technologies, one has to use the mesh topology network (figure 3.1) to achieve greater coverage. This requires more computing power (therefor more energy use), because each device also serves as a router and reroutes traffic over the network.

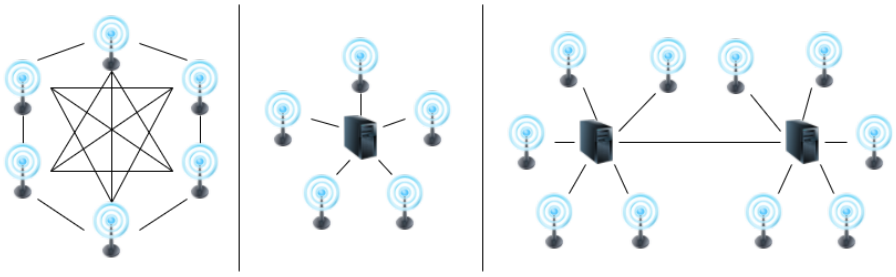

Figure 3.1: Mesh, star and star-of-star topologies from left to right, respectively.

On the other hand, LPWAN technologies have greater range and most of them use star or star-of-star topology network (figure 3.1). Such topologies consist of end-devices (sensors) and gateways (they can be computers or sensors), who are relaying messages from end-devices to the central network server. End-devices use single-hop wireless communication to send data to one or many gateways, while gateways are connected to the network server via standard IP connection. Advantages of this topology are lower costs, less processing power and less traffic overhead.

These days almost every LPWAN uses cellular networks and cheap GPRS modems but as these disappear and 3G or 4G devices are more expensive, cheaper alternatives such as LoRaWAN, Sigfox or Weightless can be adopted.

3.2 Most popular LPWAN technologies

This section shortly describes today's most popular LPWAN/IoT technologies and presents their pros and cons.

3.2.1 LTE-M

Unlike other IoT technologies where devices connect to the base stations, devices using LTE-M can connect directly to 4G network instead. With that in mind LTE-Ms advantage is that infrastructure is potentially already there and deployment strategies are well established. It also has the highest throughput of all IoT technologies. Its downsides are higher hardware price and dependency on mobile network operators.

3.2.2 LoRaWAN

LoRaWAN is open standard, standardized by LoRa Alliance, which uses LoRa technology developed and licensed by Semtech. LoRaWAN has large and influential members like Cisco and IBM. It uses the narrow band sub-GHz spectrum, has great (AES CCM (128-bit)) security and devices using it can run on batteries for several years [2]. Its downsides are ALOHA-type protocol, which makes acknowledgement difficult, limited downlink capability and not fully open standard.

3.2.3 SigFox

SigFox is most widely spread in Europe. Messages are transmitted using Ultra Narrow Band modulation (100 Hz wide band) and have bit rate of 100 or 600 bits per second [35]. Due to long messages air time, the architecture in the USA has to be significantly different than the one in Europe [36]. It is power efficient and great for simple monitoring, but it is not an open protocol and is limited to SigFox networks only. Its downside is also a minimal built-in security (16 bit encryption) and slow downlink connection.

3.2.4 LinkLab

Link Lab is LoRa Alliance member and uses LoRa on physical layer. However, instead of LoRaWAN, it uses a proprietary software on MAC layer called Symphony Link. In contrast to LoRaWAN, Symphony Link has some additional features built in like bi-directional message acknowledgment, capabilities of using repeaters, over-the-air firmware upgrades and no duty cycle limits [37]. Its downside is the dependency on the Symphony Link software.

3.2.5 Ingenu

Ingenu RPMA operates in 2.4 GHz band which is globally available. It has better uplink and downlink bandwidth speeds than most of other IoT technologies and supports a great amount of nodes per one AP. Ingenu also claims to have better doppler, interference and scheduling robustness. Their AP can cover more area than other LPWAN technologies APs (Ingenu RPMA uses one AP per LoRa's 18, cellular's 30, and SigFox technology's 70) [38]. Its downside is that it uses 2.4 GHz spectrum which is quite busy. There can be interference form WiFi and Bluetooth and it is not good at penetrating buildings and walls.

3.2.6 Weightless

Weightless is the only fully open-standard which operates in ultra narrow band in sub-GHz unlicensed spectrum. There are three types of Weightless open standards available: Weightless-N, Weightless-P and Weightless-W. Weightless-N supports only one way communication, it is simple, low cost and has the longest battery life. Weightless-P is high performance, it supports two way communication, but has slightly shorter range than the other two. Weightless-W also supports two way communication and uses, unlike other two, TV white space spectrum. All of them have great security (AES-128/256 encryption and authentication) [39]. The cons of these technologies are the slightly higher costs of devices and shorter battery life.

3.3 Internet of Things platforms

As the IoT is ever more popular and widespread, the need for data collecting and analysis tools is growing too. Today, there are many cloud platforms for saving and analyzing measured data exists. Comparison between a few of them is presented in table 3.1.

	Price	Supports multiple boards	Request methods	Response data format	Easy to use (simple dashboard)	Maximum number of connected devices in free price plan	Alerts and notifications	Built-in analytics	Max. API requests (per day)
www.thingworx.com	30 days trail	No. Only Raspberry Pi	REST (POST, GET, PUT)	JSON	No	n/a	Yes	Yes	n/a
thingspeak.com	Free	Yes. Arduino, Raspberry Pi and others	REST (POST, GET, PUT)	JSON, XML, web page(HTML), text	Yes	Unlimited	Yes	Yes	Unlimited
www.exosite.io	Limited free edition	Yes. Arduino, Microchip and others	REST (POST, GET)	JSON	Yes	10	Yes	No	n/a
www.carriots.com	Limited free edition	Yes. Arduino, Raspberry Pi and others	REST (POST, GET, PUT, DELETE)	JSON, XML, web page(HTML)	Yes	2	Yes	Yes	1000
grovestreams.com	Limited free edition	Yes. Arduino, Raspberry Pi and others	REST (POST, GET, PUT)	JSON	No	n/a	Yes	Yes	n/a
www.paraimpu.com	Limited free edition	Yes, Arduino, Inel Galileo and others	REST (POST, GET, PUT, DELETE)	JSON	Yes	4	Yes	No	1440 (1 per minute)
buddy.com	Free	Yes. Arduino, Raspberry Pi and others	REST (POST, GET, PUT, PATCH, DELETE)	JSON	Yes	Unlimited	Yes	Yes	1728000 (20 per second)

Table 3.1: The IoT cloud platforms (data are from September 2016).

Six out of seven platforms are free to use, but they have few limitations like a maximum number of devices connected or a limited number of API requests per day. The only exception between them is Thingspeak platform which enables the user to connect unlimited number of devices which can make an unlimited number of API requests.

Though Thingspeak meets all our criteria for this master's thesis research platform, we decided not use it. The main reason is that we wanted to be independent of any platform and to have control over the data and platform as a whole. With that in mind, we decided to build our own research/IoT platform. We chose to use MQTT protocol for communication between devices (it is lightweight and enables greater customization of our platform) and Python scripts for parsing and saving measured data to SQL database (how the research platform was build is presented in chapter 5).

3.4 Message Queue Telemetry Transport (MQTT) protocol

MQTT is a M2M/IoT connectivity protocol. It is a lightweight publish-subscribe-based protocol with a small code footprint [40]. It consists of two main entities: the broker and the client.

- **Broker (web server)** is the hearth of the publish/subscribe protocol. It is responsible for receiving all messages, filtering and sending them to subscribers. It also maintains the sessions with all connected clients.
- **Client** can be a publisher or a subscriber (or both at the same time). A MQTT client is any device running the MQTT library. It can be a simple sensor connected to the Internet or a big server. Its role is to publish the obtained data or listen (subscribe) to the brokers topic.

The topics act as the "routing" information for the messages. The device can subscribe to or publish to one or multiple topics. When the message is

published to a certain topic, the broker sends this message to all devices that are subscribed to it. For example: the wind sensor publishes data to the topic *ljubljana/wind-direction* and the desktop computer, subscribed to this topic, receives and analyzes these data.

Because this is a lightweight protocol, there is little security present. Only username and password can be used when the device subscribes to/publishes to the topic.

Chapter 4

LoRa and LoRaWAN

LoRa itself is a physical layer of an OSI model. It is partially open standard: the chips design is licensed and developed by Semtech, while the MAC layer is license free. LoRaWAN, on the other hand, is defined on the MAC layer of the OSI model and uses a subset of LoRa modulation parameter settings. It is license free and standardized by LoRa Alliance.

4.1 Lora

LoRa is the long-range, low-power, low-bitrate, wireless technology developed by Semtech, mainly targeted for evermore developing M2M and IoT networks. Due to low bit rate, devices using this technology are capable of working on a single battery for a few years and are perfect for monitoring remote locations [2].

LoRa technology uses the Chirp spread spectrum (CSS) technique to encode information which makes it resistant to Doppler effect. It also enables the use of cheap oscillators and makes the synchronization even faster. Due to its big link budget, with more than 150 dB, one can achieve good coverage.

LoRa operates in license free ISM radio bands at 434 MHz or 868 MHz in Europe and 929 MHz in the USA [25].

4.1.1 Modulation

LoRa modulation is based on the CSS scheme which uses wideband linear frequency-modulated pulses, whose frequency decreases or increases over a specific amount of time based on the encoded information [26].

Each bit of information is encoded by multiple chirps (figure 4.1). The relation between bits and LoRa chirps is given by:

$$Rc = 2^{SF} Rb \tag{4.1}$$

where SF is the spreading factor, Rc and Rb are LoRa modulation chip and bit rate, respectively.

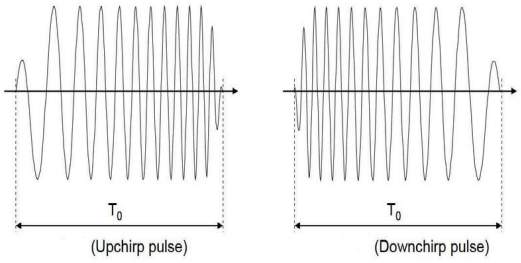

Figure 4.1: Upchirp (left) and downchirp (right) modulated linear frequency in the time domain [27].

In LoRa modulation, the spreading of the spectrum is achieved by generating a chirp signal that continuously varies in frequency. An advantage of this method is that the timing and frequency offsets between the transmitter and the receiver are equivalent, greatly reducing the complexity of the receiver design [30].

This modulated signal (a LoRa packet in particular) can bee seen in figure 4.2, which shows a captured LoRa signal in the time-frequency domain. Ten up-chirps and two down-chirps (from the left) represents a LoRa packet preamble, while the others are packet payload.

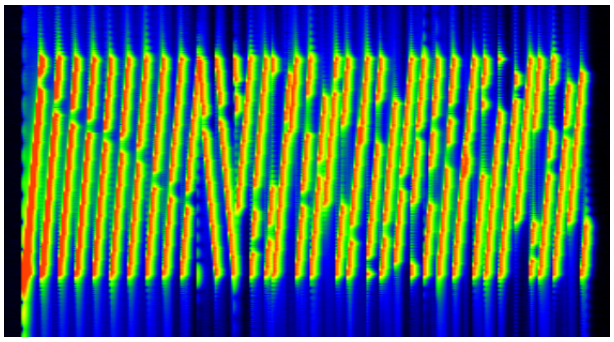

Figure 4.2: A captured LoRa signal in the time-frequency domain. The image has been captured using the CubicSDR [28] software and the RTL-SDR [29] device.

4.1.2 Modulation parameters

The following parameter values are possible to use with LoRa modulation:

Parameter	Value
Chanel bandwidth [kHz]	7.8, 10.4, 15.6, 20.8, 31.25, 41.7, 62.5, 125, 250, 500
Spreading factor	6, 7, 8, 9, 10, 11, 12
Coding rate	1, 2, 3, 4

Table 4.1: All of the possible values for the LoRa modulation.

Both range and bitrate can be controlled by choosing a different combination of parameters. For example: choosing a higher spreading factor and a lower bit rate leads to an increased range, while choosing a higher bit rate and a lower spreading factor increases the bit rate but lowers the range.

4.1.3 Packet structure

There are two types of LoRa packet format, the explicit one and implicit one.

The explicit packet format includes a header which contains information about the number of bytes sent in the packet, its payload coding rate and whether a CRC is used (figure 4.3). When using this format, one may set

modulation parameters only on the transmitter side, while they are automatically obtained from the header on the receiver side. On the other hand, if the implicit packet format is used, the payload size is calculated from the packet payload, while the coding rate and the CRC have to be set both on the transmitter and the receiver. If the application is transmission intensive, an implicit header mode can be used to reduce the packet air time.

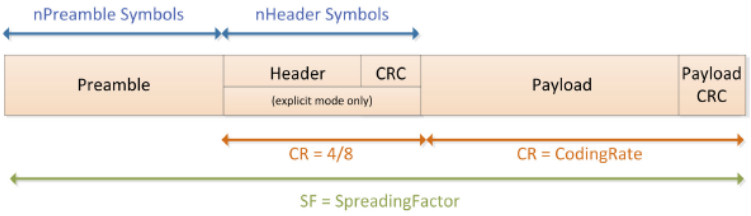

Figure 4.3: A LoRa packet structure [2].

Preamble

The preamble is used to synchronize the receiver with the incoming data flow. Its length is programmable and can be set from 10 to 65539 symbols, however the minimum length is sufficient for all communication. The preamble length should be configured identically on the transmitter and receiver side. If it is not known, the maximum length should be set on the receiver side [2].

Header

As mentioned earlier, two types of header mode are available: the implicit one and the explicit one.

The explicit header provides information about the payload length in bytes, the forward error correction code rate and the presence of the optional 16-bit CRC for the payload (figure 4.3). It is transmitted with the error correction code of 4/8 and has its own CRC to allow the receiver to validate the received headers.

Implicit header mode does not contain the header and its information about the payload length, the forward error correction code rate and the presence of the payload CRC. If an implicit header mode is used, these parameters have to be manually set on the transmitter and the receiver side. The advantage of the implicit header mode lies in the packet transmission time reduction.

Payload

A packet payload contains actual data that is sent over the air. This data is coded as specified in the headers parameter error-correction code rate, if the packet has an explicit header, or read from the registers otherwise.

Packet on-air time

The total on-the-air transmission time of the LoRa packet can be calculated using the spreading factor (SF), the coding rate (CR) and the signal bandwidth (BW) as follows.

Symbol duration *Tsym* is defined as:

$$Tsym = \frac{2^{SF}}{BW} \tag{4.2}$$

With *Tsym* known, the preamble duration *Tpreamble* can be calculated:

$$Tpreamble = (n_{preamble} + 4.25)Tsym \tag{4.3}$$

where $n_{preamble}$ is the number of the preamble symbols.
The total number of symbols that makes the packet is defined by:

$$payloadSybNb = 8 + max(\lceil \frac{8PL - 4SF + 28 + 16 - 20H}{4(SF - 2DE)}(CR + 4)\rceil, 0) \tag{4.4}$$

where the

- PL is the number of payload bytes (1 to 255),
- SF is the spreading factor (6 to 12),
- H = 0 when the header is enabled and H = 1 when no header is present,
- DE = 1 when the low data rate optimization is enabled , DE = 0 for disabled,
- CR is the coding rate from 1 to 4 (1 corresponding to 4/5, 4 to 4/8) [31].

The payload duration can be calculated by multiplying the total number of symbols with $Tsym$:

$$Tpayload = payloadSybNb * Tsym \tag{4.5}$$

From the equations 4.3 and 4.5, the total LoRa packet time on air can be calculated:

$$Tpacket = Tpreamble + Tpayload \tag{4.6}$$

4.2 LoRaWAN

LoRaWAN is a LPWAN specification intended for wireless battery operated Things in a regional, national or global network [2]. As is typical for LPWANs, LoRaWAN network arhitecture is laid out of star or star-of-star topology. Its end-devices use single hop LoRa or FSK communication. Every end-point communication is generally bi-directional, but also supports the multicast to reduce on air time and is spread out on different frequencies channels and data rates. Due to the spered spectrum technology, communications with different data rates do not interfere with each other. This leads to an increased capacity of the gateways. LoRaWAN data rates range from 0.3 kbps to 50 kbps, while their selection is a tradeoff between communication range and message duration [33].

4.2.1 LoRaWAN network classes

LoRaWAN network distinguishes between three types of end-devices: Class A, Class B and Class C. Each end-device connected to LoRaWAN network has to implement at least Class A functionality, the other two are optional.

- **Bi-directional end-devices (Class A):** Devices of this type allow bi-directional communication. Each up-link transmission is followed by two short receive windows. The transmission slot has a small variation in time based on application needs. As a consequence to these time variations, receive windows times also vary. This type of operation has the lowest overall power consumption and is ideal for the application that only requires downlink communication from the server shortly after uplink transmission. If downlink communication happens outside the receive windows, device will ignore it or even not detect it, if it is in the sleep mode.

- **Bi-directional end-devices with scheduled receive slots (Class B):** In addition to Class A random receive windows, Class B end-devices also allow an extra received window at the scheduled time. In order to open the synchronized receive window, the gateway has to send the time synchronized beacon to the end-device.

- **Bi-directional end-devices with maximal receive slots (Class C):** End-devices of this type have a continuously opened receive window, when they are not transmitting. They use the most power, but offer the lowest latency for server to end-device communication. This is also the class, as will be described later, which is used in our platform.

4.2.2 LoRaWAN modulation parameters

Unlike LoRa, LoRaWAN uses only the following subset of LoRa modulation parameters:

- LoRa modulation with 125 kHz bandwidth and spreading factors from 7 to 12 or
- LoRa modulation with 250 kHz bandwidth and spreading factor 7 only.

LoRa packets have to have both an explicit header and an enabled CRC and they have to use 4/5 coding rate [34].

In LoRaWAN, the Gaussian frequency-shift keying (GFSK) can also be used for a high speed uplink channel. It has to use the frequency deviation of ± 25 kHz and 50 kbit/s bit rate.

Chapter 5

Our test platform

Our test platform (figure 5.1) is the heart of this master's thesis. It was designed to be flexible, easy to use and with scalability in mind. It supports both long term tests with stationary end-devices and short term tests with moving (mobile) end-devices, who can all be preformed indoor or outdoor.

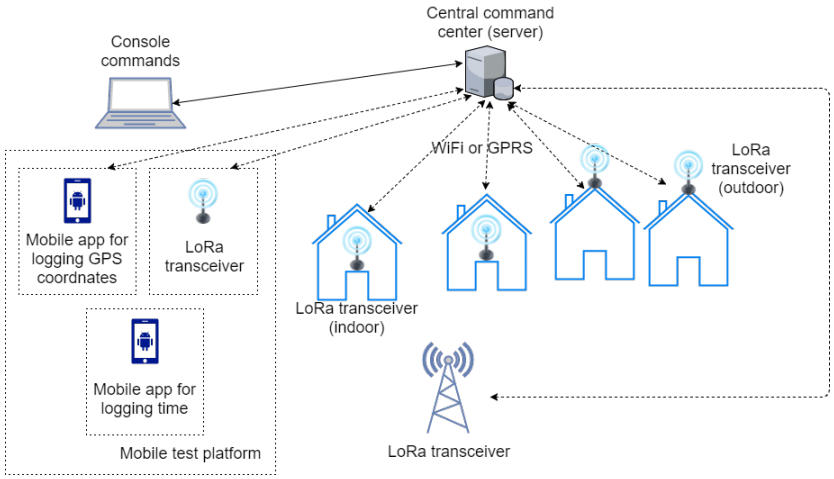

Figure 5.1: An overview of our research platform as a whole.

This platform consists of gateways, end-devices and Android applications which are all connected to the central command center (server) via IP connection. This server controls all end-devices and gateways and stores measured data in the SQL database. All necessary settings on the end-devices and

gateways can be changed by sending one JSON file with a simple terminal command (see listing 5.1) to MQTTs topic named *lora/test*.

With this test platform one can not only test the performance of the LoRaWAN, but LoRa in general. The tests can be performed with all possible combinations of LoRa parameters listed in table 4.1. If LoRaWAN is subject to the test, one only has to set the appropriate parameters and the test can begin. Unfortunately, the FSK modulated LoRaWAN uplinks can not be tested, because this type of modulation is not supported, but can be enabled with a few minor modifications.

When one performs the outdoor tests with the moving end-devices, a GPS position has to be measured and logged. For this purpose, the Android application has been developed. It measures the GPS position and sends it to the server where the data is stored. Because GPS does not work indoor, another Android application for logging the indoor "position" had to be developed.

This chapter describes our test platform in details, how it is made and how does it works.

5.1 Gateways and end-devices

Gateways and end-devices are one of the most important things in the IoT field. Without them, one could not gather, send or receive data.

This sub chapter describes the hardware used in our test platform and the software running on it. It also presents three different operation modes of the LoRa end-devices and gateways that can be used in our research platform.

5.1.1 Hardware

Raspberry Pi 3 Model B:

For our test platform, we decided to use Raspberry Pi 3 Model B for gateways and end-devices. They are easy to use and relatively cheap, but most

important, one can connect a few different modules and sensors to them. This model comes with the integrated WiFi and Bluetooth modems which ease the use of the test platform and allows the device to receive commands from the central command center or to send measured data to the server, regardless whether there is the LoRa connectivity or not.

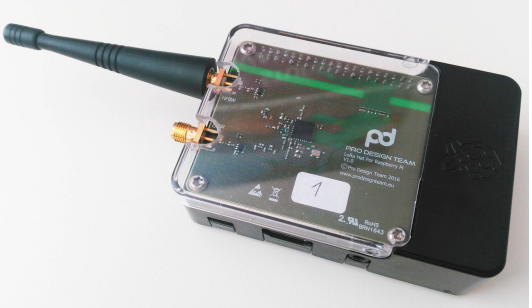

Figure 5.2: Our Raspberry Pi 3 Model B with a custom LoRa module and a 868 MHz antenna connected to a high-frequency connector.

LoRa module for Raspberry Pi 3

On the top of the Raspberry Pis 3 we put the custom made LoRa module (figure 5.2) with the Semtech SX1276 transceiver which communicates with Raspberry Pi 3 over the SPI bus. This setup enables us to conduct tests both in 433 MHz (low frequency) or 868 MHz (high frequency) frequency bands. The switching between high and low frequencies can be done programatically on the fly during tests. However, we never had both antennas connected at the same time and the tests on how this effects the reception were not conducted. Within this work, we only used the 868 MHz frequency band and the 1/4 wave whip antenna.

5.1.2 Software

LoRa chip drivers

Before we were able to use our chip, we had to modify the Semtechs SX1276 drivers, written in C programming language. The modifications had to be made to the Digital I/O (DIO) pin layout which is a little different on our module than it is in the drivers.

LoRaC program

For the needs of our testing platform, we took the "ping pong" program that comes with the Semtech SX1276 drivers, and modified it (we will call this modified "ping pong" program LoRaC). Its work flow is shown in figure 5.3.

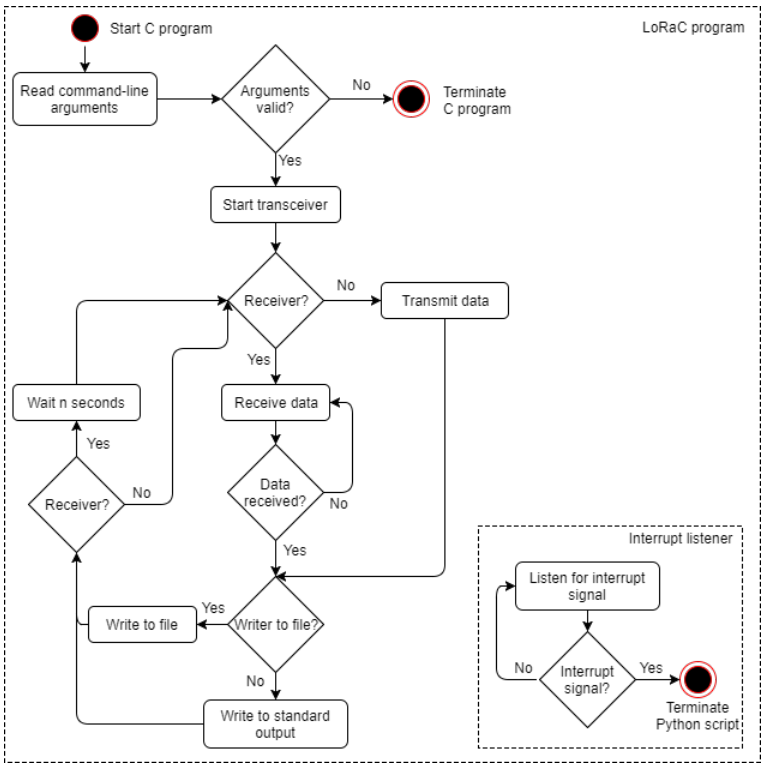

Figure 5.3: The LoRaC program workflow.

This program does not expect any inputs while it is running. All the

necessary settings and the commands have to be provided with command line arguments at the program start-up. These commands are:

- **-r**: start the device in a receiver mode. If this flag is not present, the transmission mode is activated. If *-bi* argument is also present, *-r* only indicates whether the program starts as the receiver and switches to the transmitter after the packet airtime + T millisecond (where T is the value set by *-d* parameter at the program start-up), or vice versa.
- **-f**: if present, the program writes measurements to the file, otherwise the measurements will be written to the standard output instead.
- **-d**=T: sets the T milliseconds delay between the airtime of two packets.
- **-b**=N: sets the LoRa bandwidth value where N can have values from 1 to 9. N represents the bandwidth values as follows: 0: 7.8kHz, 1: 10.4 kHz, 2: 15.6 kHz, 3: 20.8 kHz, 4: 31.2 kHz, 5: 41.6 kHz, 6: 62.5 kHz, 7: 125 kHz, 8: 250 kHz, 9: 500 kHz.
- **-s**=N: sets the spreading factor, where N can be one of the values between 6 and 9.
- **-c**: if this flag is set, CRC is turned on.
- **-e**=N: sets the error coding rate (CRC). Possible values are 1: 4/5, 2: 4/6, 3: 4/7, 4: 4/8.
- **-E**: this flag turns the explicit header on. The default is off.
- **-lf**: when this flag is set, the transceiver operates on 434 MHz. If it is not present, transceiver frequency is set to 868 MHz.
- **-bi**: when this option is set, the transceiver alternates between the transmitter and the receiver. For a detailed explanation, see section 5.1.3.

If all the command line arguments are correct, LoRaC starts transmitting or receiving data. If not, the program is terminated and the user is notified about the wrong command line arguments.

When data is transmitted or received, LoRaC writes all relevant information to the standard output or to the file, if *-f* argument was provided at the programs start. This includes the information about the received/transmitted packet status, the status of the program itself or general information. The decision to write all this to the standard output was made because, as we will describe later, LoRaC runs as a subprocess inside the Python script where its outputs are than parsed.

The information about the transmitted packet is, as shown in the program outputs 5.1 - 5.3, *packet status*, *packet payload* and *epoch timestamp with accuracy of microseconds* when the packet was sent, from left to right, respectively. The packet status is always OK, since it is there just for compatibility with the received packet output structure. The individual values are separated with a colon and a semicolon for easier parsing.

$$OK : 1; 1498634655.458129; \tag{5.1}$$

$$OK : 2; 1498634658.471680; \tag{5.2}$$

$$OK : 3; 1498634661.484587; \tag{5.3}$$

On the receiver side, LoRaC prints statuses of the Semtechs SX1276 modem (see program outputs 5.4 - 5.6 and 5.8 - 5.10) and the information about received packet (program outputs 5.7 and 5.11). These statuses represent the current SX1276 modem state and consequently the packet status. They can be accessed in the modems register named *RegModemStat* at the address *0x18*. There are five different modem statuses, but only the following are relevant for our platform:

- **rxOnGoing** means that the chips transceiver is in reception mode and listens for potential ongoing packet transmission,

- **signalDetected** event occurs when a packet (concretely a packets preamble) is detected and
- **headerInfoValid** is triggered when and only if the packet header CRC is valid.

$$rxOnGoing : 1498634652.251416; \tag{5.4}$$
$$signalDetected : 1498634655.420648; \tag{5.5}$$
$$headerInfoValid : 1498634655.422046; \tag{5.6}$$
$$OK : 1; -104.0; -78.1; 8; 1498634655.432040; \tag{5.7}$$
$$rxOnGoing : 1498634655.432155; \tag{5.8}$$
$$signalDetected : 1498634658.433948; \tag{5.9}$$
$$headerInfoValid : 1498634658.435429; \tag{5.10}$$
$$OK : 2; -98.0; -52.5; 9; 1498634658.445422; \tag{5.11}$$

When the packet is received, the program output (5.7 and 5.11) contains the information about *packet status*, *packet payload* (in our case transmitted packet sequence number), *average packet RSSI*, *current RSSI*, *packet SNR* and *epoch timestamp* when the packet was received, from left to right, respectively. The values of this outputs are also separated with a colon and a semicolon for easier parsing. The received packet status can be:

- **OK:** the packet has been received and its calculated CRC matches the one in the packet, or
- **CRCERR:** when the calculated packets CRC do and packets CRC does not match.

Error and info messages, written to the standard output of LoRaC program, are mainly associated with the users input and its validation.

LoRaC also contains a signal interrupt listener, which listens for the combination of *Ctrl + c* keys and correctly terminates the program when this combination is pressed.

Python script

The gateways and the end-devices "brains" lie in the Python script (lets call it LoRaPython). This script listens to the commands sent over the MQTT (is subscribed to the topic), runs the LoRaC program in subprocess, parses its outputs and publishes (sends) them to the main server.

When one wants to run this script all the necessary settings have to be provided as the command line arguments. It accepts the same arguments as LoRaC does, described in previous section, with one additional command: **-start**. If this command is present, the transceiver is started immediately after the program starts and does not wait for the start command from the server.

As we can see from the figure 5.4, when scripts starts, the settings for each individual device are read from JSON file. This settings are (as shown below in the example of JSON file) *device id*, MQTT brokers *login credentials*, MQTT brokers *IP adress*, *port* and *keepAlive* which defines the maximum period in seconds allowed between the communications with the broker. If no other messages are being exchanged, this controls the rate at which the client will send ping messages to the broker [41].

```
{
        "id":1,
        "username": "username",
        "passwd": "password",
        "ip": "192.168.1.100",
        "port": 8883,
        "keepAlive": 60
}
```

Next, the script reads the command line arguments and validates them. If they are not valid, the script terminates and warns the user about the wrong inputs. When all of the provided arguments are valid, script runs two parallel threads; one with the MQTT client and one with the LoRaC subprocess. Parallel threads are needed because, both the MQTT client (the paho-mqtt library [41]) and the part of the script where LoRaC runs, are

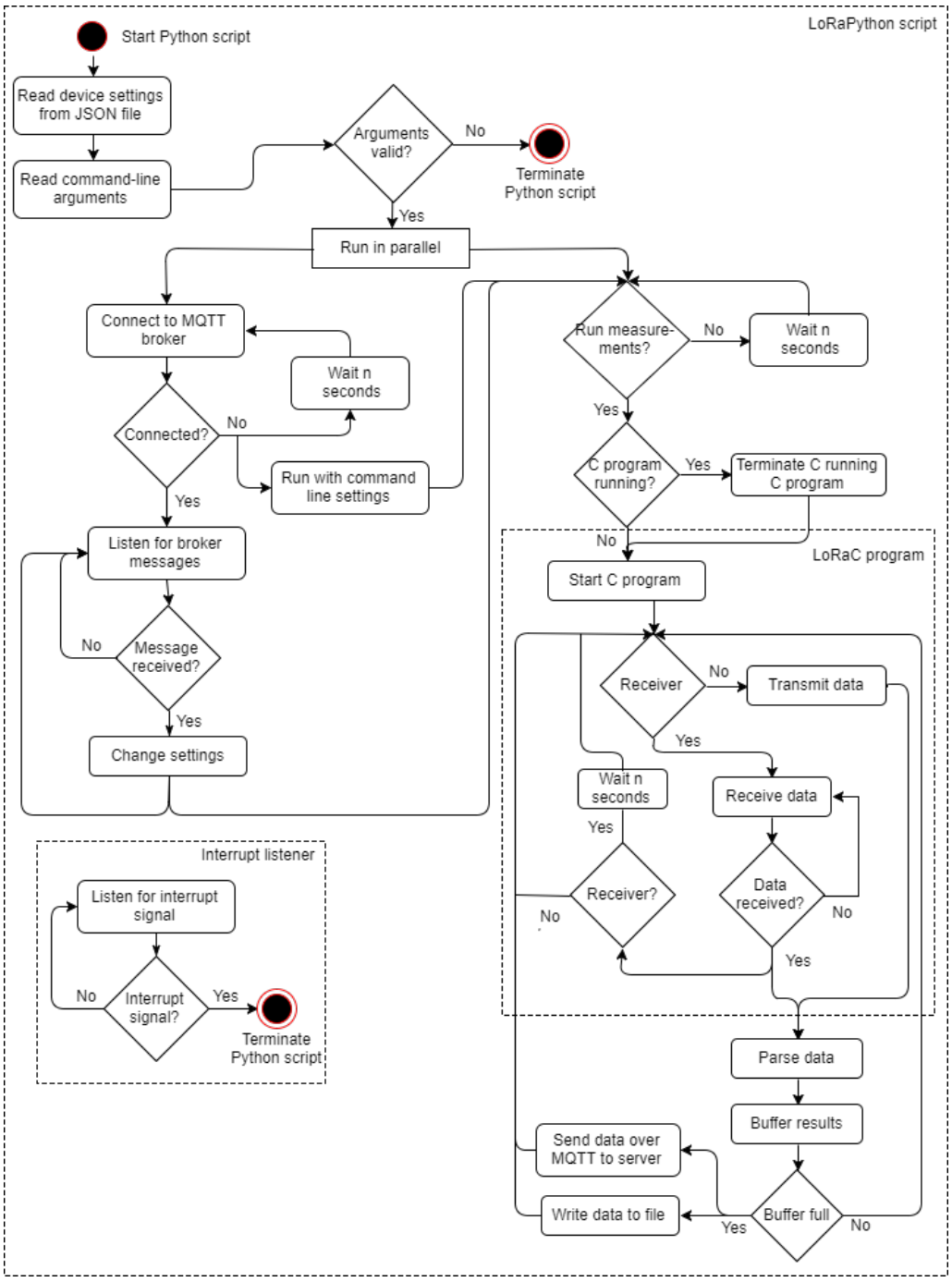

Figure 5.4: The workflow of LoRaPython and LoRaC running on the gateways and end-devices.

blocking parts of the code. Although the paho-mqtt library has the option to run in the non-blocking mode, we decided not to use it, because it does not present a big difference in implementation.

Threads workflows are as follows:

- **MQTT client thread**: when started, it tries to connect to the MQTT broker, using the settings from the file. If the device is unable to connect to the broker, because there is no connection or a connection timeout occurred, LoRaC subprocess is automatically started and does not wait for the server commands. We implemented this automatic subprocess start to cover the case when there is no internet connection on the location, where tests are being conducted. In this case all necessary settings must be provided with the command line arguments at the program start-up. Meanwhile, MQTT client tries to connect to the broker every 30 seconds, just in case if, the device connects to the Internet.

  When the client establishes a connection with the broker, it starts listening for messages. They contain all of the necessary instructions and settings for running the measurements. These messages have the following JSON structure:

```
{
"devices": [{
        "id":1, // device id
        "rec": true, // if true, device is receiver,
                // otherwise transmitter
        "bidirectional": false, //if set to true,
                // device alternates between
                // receiver and transmitter
        "setId": 1, // settingId (from JSON array
                // "parameters") to use with this device
        "start": true, // start measurements
        "restartT": true, // restart LoRaC subprocess
        "debug":true,
        "gpsLat":46.050343, // device GPS latitude
```

```
        "gpsLon": 14.469144, // device GPS longitude
        "locDesc":"Test location" // device descriptions
}],
"buffer":{"buffSize": 32}, // number of LoRaC output lines
                           // buffered before they are send to
                           // MQTT broker
"parameters": [{
        "settingId": 1, // setting id
        "bw": "8", // bandwidth
        "sf": "6", // spreading factor
        "crcOn": true, // is CRC on
        "crcVal": "2", // coding rate value
        "delay": "3000", // delay between packets airtime
        "lowFreq": false // if true, transceiver operates
                         // on 434 MHz, otherwise on 868 MHz
}]
}
```

This example message contains settings for only one device (object in *devices* array). There can be multiple objects presented in *devices* and *parameters* JSON arrays. *Devices* array has to have one JSON object for each individual device one wants to control, on the other hand, multiple devices can use one object from the *parameters* array.

- **LoRaC parser thread**: if the message is received and *restartT* (restart transmission) and *start* (start transceiver) parameters for the particular device are set to true, then the script starts new a subprocess with LoRaC program on this device. If another subprocess is already running, script first terminates it and the starts a new one (this is because LoRaCs settings can not be changed during its runtime).

  If one wants to terminate the subprocess without starting a new one (end the measurements), the *restartT* parameter has to be set to true and the *start* parameter to false. If the *restartT* parameter is set to false, the script will not change the settings nor will it start anew or restart running the LoRaC subprocess, regardless the value of the *start*

parameter. This is because message with instructions is broadcasted to all of the connected devices and if one does not want to restart measurements on a particular device(s), the *restartT* parameter has to be set to false.

When LoRaC writes results to the standard output, LoRaPython catches, parses and buffers them. When the buffer is full (its size is specified in the received instruction messages parameter *buffer*), the results are converted to JSON message and published to the broker (MQTTs topics name is in our case *lora/testPub*). Script "publishes" the results even if the MQTT client is not connected to the broker. In this case, the paho-mqtt library buffers messages in its own buffer and publishes them all together when a connection with the broker is established.

All of the parsed results are, as a backup, saved to the CSV file too, just in case the MQTT client does not connect to the server before the device is powered off.

LoRaPython also has the signal interrupt listener and listens to *Ctrl + c* keys combination. When these keys are pressed, LoRaPython sends the SIGINT signal to the LoRaC subprocess to terminate it. After the subprocess is terminated, LoRaPython publishes all of the previously unpublished buffered results, disconnects the MQTT client from the broker and finally terminates itself.

5.1.3 Operation modes

Our gateways and end-devices can operate in three different modes:

- **Transmitter mode:** in this mode, the device can only transmit data. Delay between the airtime of two packets is set by the *-d* command line parameter at the scripts startup. For an easier data analysis, the transmitter is sending (as a packet payload), a sequence number of the transmitted packet.

- **Receiver mode:** the device constantly listens for potential packets on the channel, parses and publishes them to the MQTTs topic.
- **Bi-directional mode:** this is a special kind of the operation mode. It consists of the alternating transmitter and the receiver modes. This case was introduced to out platform for measurements where one wants to test how the signal propagates in both directions with no need for having two more extra devices (the uplink and downlink communication).

Figure 5.5: An example of bi-directional communication between devices A and B. The solid line represents the packet traveling direction in the first iteration.

An example of how this bi-directional mode works, is shown in figure 5.5, where device A is in the first iteration transmitter which transmits a packet and device B acts as the receiver and receives this packet. In the next iteration, the devices change roles (device A is now the receiver and device B is the transmitter) and so on.

It is very important that two devices, which communicates in bi-directional mode, maintain synchronization. If they lose it, it could happen that they could be, after some time, in the transmitter or receiver mode at the same time. To prevent this from happening, we implemented switching between the modes and synchronization between the two devices (figure 5.6) as follows:

- if the device is in the transmitter mode, it transmits the packet and immediately after the transmission is done, it switches to the receiver mode,

– if the device is in the receiver mode, and it has just been switched on (first the transmit/receive iteration), then the receiver listens for *airtime + delayBetweenAirTime* milliseconds for potential packets, where *airtime* is the time that the packet needs to travel over the air and *delayBetweenAirTime* is the delay between the airtimes of two packets designated by the *-d* command line argument at the program start. If no packets have been received during this time, then the device switches to the transmitter mode. If the packet has been received, the receiver waits for *delayBetweenAirTime* milliseconds after the packet arrival, before switching to the transmitter mode.

  On the other hand, if this is not the first transmit/receive iteration than the receiver waits for *airtime + 2 * delayBetweenAirTime* milliseconds for the potential arrival of the packet before switching modes. This covers the case when the packet had been transmitted and the receiver did not get it. In case the packet was received during this time, the receiver waits for *delayBetweenAirTime* milliseconds after the arrival of the packet and then switches to the transmitter mode.

  How the *packet airtime* can be calculated is shown in chapter 4.1.3.

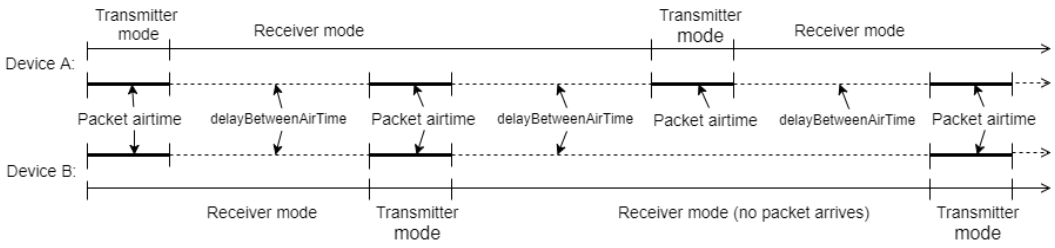

Figure 5.6: Synchronization between two devices using received packets.

Although this synchronization should work in theory work, it seems that to make it work in reality is not that simple. In our research

platform this works perfectly with the *delayBetweenAirTime* of 3000 milliseconds or greater. If the *delayBetweenAirTime* is set to 2500 millisecond it looses synchronization every few packets, with the delay lower than that when it does not work. We did not research why is this happens because, for our needs, the *delayBetweenAirTime* greater than 3000 milliseconds was enough. Further work remains to be done regarding this research.

5.2 GPS tracking Android application

When one wants to conduct measurements which includes moving nodes, these nodes have to have some kind of GPS tracking which is needed for the later data visualization. A different kind of GPS receivers for Raspberry Pi 3 exists today on the market, but the good ones are quite expensive.

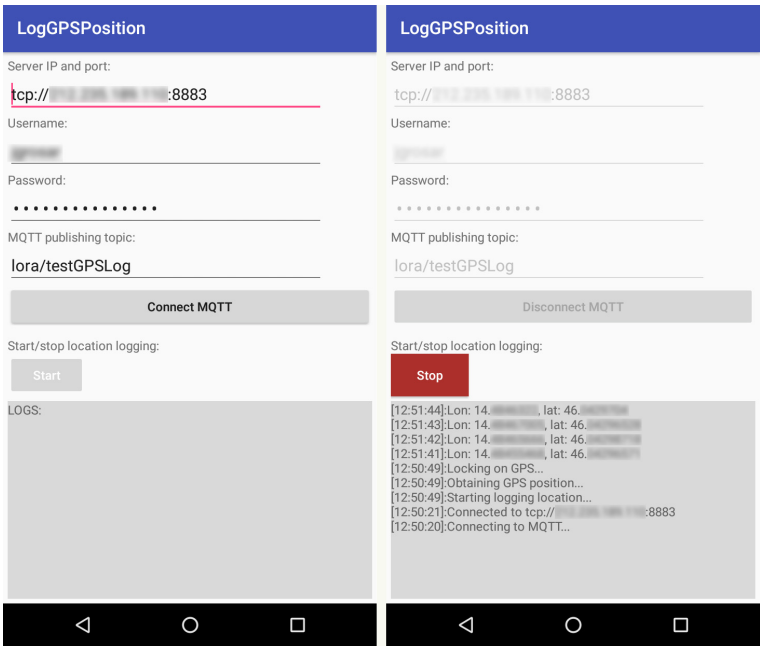

Figure 5.7: Android application for GPS logging.

To cut the costs of the research platform, we decided to a create simple

GPS location logging application for Android mobile phones, called LogGPSPosition. This application is intended for outdoor use, where the GPS signal is available.

On the application start-up, one has to connect to the MQTT broker first (left app screenshot in figure 5.7). When the connection is established, the GPS coordinates logging can begin (right app screenshot in figure 5.7). App measures the GPS coordinates every second and publishes them to the MQTT topic provided in the input field "MQTT publishing topic".

5.3 Android application for indoor "tracking"

As one can also conduct indoor measurements, using the moving nodes, the application for logging indoor "location" has also been created (figure 5.8). In reality, it does not log any kind of location, but it does log time when the user presses the "Log time now" button. For example: the user determinates a route through the building where the moving node will move and the orientation points on it before the measurements begin. When the moving node reaches the orientation point, while moving on the predetermined route, the user logs time. By knowing the orientation points and time when the node has been on this points, one can draw a map of signal coverage in the building.

5.4 Web server

The second most important device of our research platform, besides the enddevices, is a web server where the MQTT broker and the multiple Python scripts live. These scripts listen to the MQTTs topics, receive the messages, parsing and saving them to the database. The MQTT broker allows the devices to create topics, subscribe to them and publish the messages. For our test platform, we use four different topics: *lora/test, lora/testPub, lora/testGPSLog* and *lora/testTimeLog*.

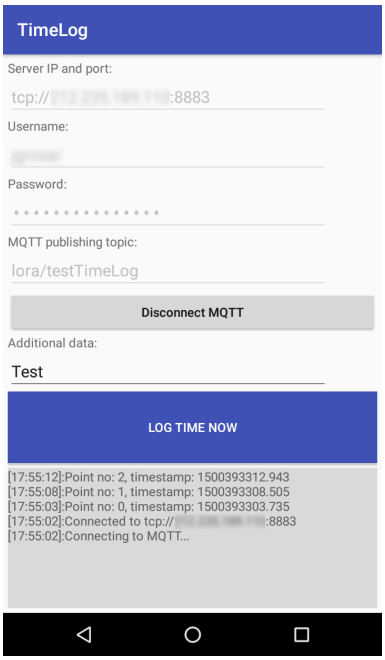

Figure 5.8: An Android application for time logging.

The *lora/test* is a special topic for broadcasting the settings (JSON file) to all of the gateways and end-devices (example of this JSON file is presented in section 5.1.2). All devices included in the research platform, must be connected to this topic for easier control over the measurements. This enables us to control them with only one command executed the on single terminal. When one wants to change the device settings (publish JSON file), the Unix command from listing 5.1 (or an appropriate command for other platforms) has to be executed:

Listing 5.1: "Unix terminal command"

```
& mosquitto_pub -h <host_IP> -p <port_number> -t "<MQTTs_topic>"
    -u <username> -P <password> -f <path_to_JSON_file>.
```

The broker then broadcasts this file to all of the subscribed devices.

On this central server, three very important Python scripts also live:

- **Script for parsing LoRa messages:** this script is subscribed to the

lora/testPub topic. The end-devices and gateways use this topic for publishing messages to the broker as described in section 5.1.2. These messages contain modem statuses and data about the transmitted and received LoRa packets.

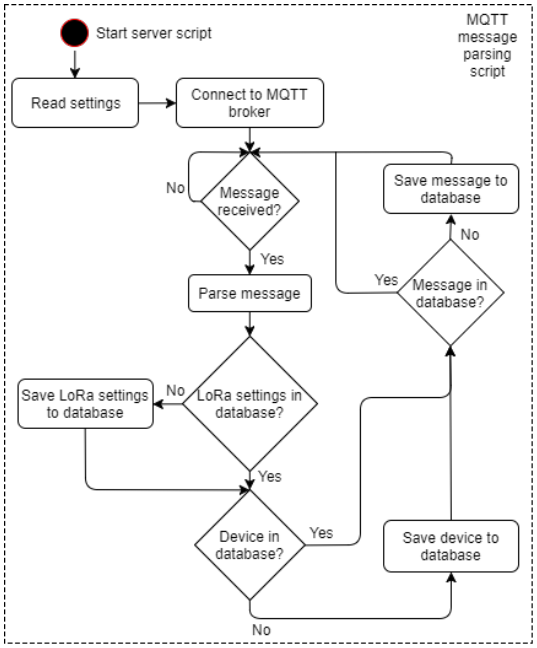

Figure 5.9: The workflow of the message parsing script.

This script first, as is the case with all of the other scripts, reads the settings from the JSON file, then connects to the MQTTs topic and waits for messages. When a message is received (see the example below), the script first looks up at the database table *settings*, if this set of the device settings (*settings* object in received JSON message) is already saved there and gets its row id. If there are no results, it saves a new set of settings to the database and gets the newly saved row id. This row id is then used for connecting *settings* and *sensor* database tables. The similar procedure happens for the existing sensor lookup in the database table *sensor*. When the sensor row id is obtained, each individual measurement (JSON object in *data* JSON array, containing

transmitted/received message data or modem status) can be saved to the database table *measurements* along with the sensor id. The example of a received JSON message with measurement results is shown below:

```
{
"devId": 1,
"location": {
        "lat": 46.049,
        "lon": 14.46101,
        "locDesc": "Bidiredctional mobile node"
}, "settings": { // device settings
        "bw": 7,
        "sf": 10,
        "crcOn": true,
        "cr": 2,
        "lowFreq": false
},"bidirectional":true,//set to true if bi-directional
                //mode has been used, false otherwise
"rec": true, // set to true if device was receiver,
                //false otherwise
"data": [{ //transmitted/received packets and modem statuses
                "status": "rxOnGoing", // modem status
                "time": 1500401309.261010
        },{
                "status": "OK", // transmitted packet
                "rxId": 393,
                "time": 1500401309.525943
        },{
                "status": "signalDetected", // modem status
                "time": 1500401312.571728
        },{
                "status": "headerInfoValid", // modem status
                "time": 1500401312.636371
        },{
                "status": "OK", // received packet
                "rxId": 394,
                "cRSSI": -110.0,
```

```
                "pRSSI": -50.3,
                "SNR": 12,
                "time": 1500401312.802434
        }]
}
```

This scripts also enables the setting up of email notifications, if some error occurs during message parsing.

- **Script for parsing GPS data from Android application:** this script is subscribed to the *lora/testGPSLog* topic and waits for the messages from Android GPS logging application. It parses the data and saves it to the database table called *travel_location*.

- **Script for parsing indoor location messages from Android application:** listens for messages coming from *lora/testTimeLog* topic and saves them to database table *point_time*.

5.5 Database

Our measurements would be useless, if we did not have the database to save measured data to. In our research platform the SQL database and the MySQL relational database management system are used.

The research platforms database consists of 10 tables (figure 5.10). The main data table, where the measured data is stored, is called *measurements*. It contains columns for all of the measured parameters: device id (*sensor_id*), packet id (*rx_id*), current RSSI (*current_rssi*), packet RSSI (*packer_rssi*), packet SNR (*snr*), epoch timestamp when the packet was transmitted or received (*timestamp*), sequential number of CRC error (if there is one) (*crc_err_no*), GPS coordinates where the packet was received (*packet_gsp_longitude* and
packet_gsp_latitude), user defined text data (*additional_sensor_data*) and packet status (*status*).

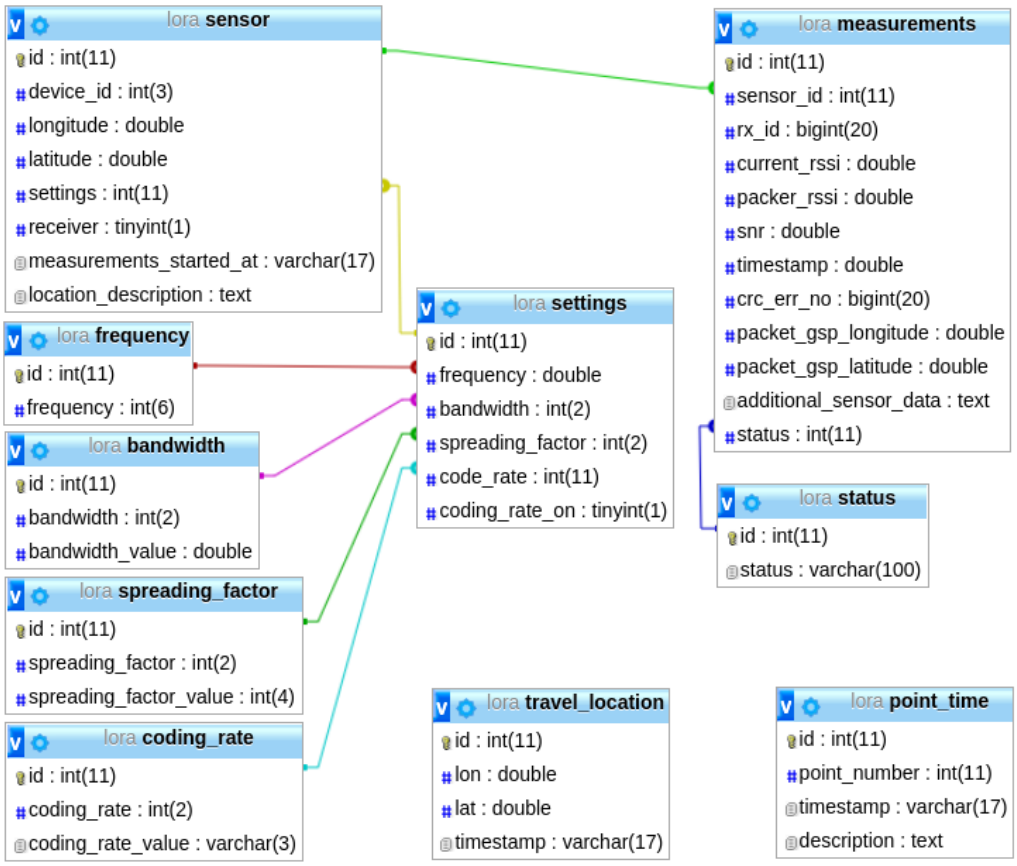

Figure 5.10: A research platforms database relational model.

The other two very important database tables are *sensor* and *settings*. In the first one, the meta data about each individual device (sensor) can be found, while in the second one, the sensor settings (frequency, bandwidth, spreading factor, coding rate and flag if CRC is on) are stored. Tables *frequency*, *bandwidth*, *spreading_factor*, *coding_rate* and *status* contains all of the possible values for LoRa frequencies, bandwidths, spreading factors, coding rates and statuses, respectively. Packet and modem status enumerators are stored in the table *status*. These statuses are OK, CRCERROR, ERROR, rxOnGoing, signalDetected and headerInfoValid.

The *travel_location* table is used for saving the GPS coordinates of moving mobile nodes captured with the LogGPSPosition Android application, while the data from the TimeLog Android application is stored in the database table called *point_time*.

Chapter 6

Measurements and results

6.1 Preliminary tests

During the development of this platform we performed several tests and measurements.When the platform seemed to be mature enough for testing, we conducted a few range tests. For all of these tests we put the transmitter (Raspberry PI 3 Model B) on some fixed location and drove, with the receiver (also Raspberry PI 3 Model B) placed on the car roof (figure 6.1), away from it.

Figure 6.1: The receiver (the Raspberry PI 3 Model B) placed on the car roof (the rotary light was used just for the easier fixing of the receiver and power bank to the roof).

6.1.1 Test in Ljubljana

The urban range test with the moving node was conducted in Ljubljana where the transmitter was placed on the window shelf on the sixth floor. Because at the time the research platform was still in the early development stage and the Android application for logging the GPS position had not been jet developed, we do not have the accurate data of the mobile node movement. The best we can do is to show the route of the moving node on Google Maps (figure 6.2).

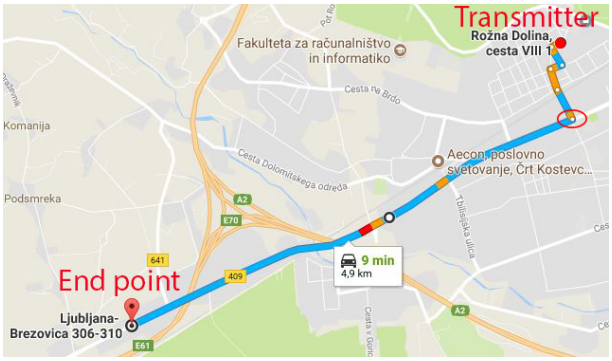

Figure 6.2: The transmitter placement, the final receiver point and the route where the receiver moved during the measurement.

Figure 6.3: A 3D map of Ljubljana showing the receiver on the furthest point from the transmitter.

For this test we used the following LoRa parameter settings: 125 kHz bandwidth, spreading factor 12, CRC on and coding rate 4/8 which gave us the packet air-time of 925 milliseconds and the bitrate of 183 bps. As figures 6.2 and 6.3 show, quite a long signal range in urban area (with no line of sight) was achieved. The air distance from the transmitter to the last point, where the signal was still received, was 4.1 kilometers despite the buildings and the slight elevation in between. At this point, the last received packet RSSI value was -128 dBm and SNR value of -18. We continued the measurement for a few hundred meters, but from this point on no packet was received anymore.

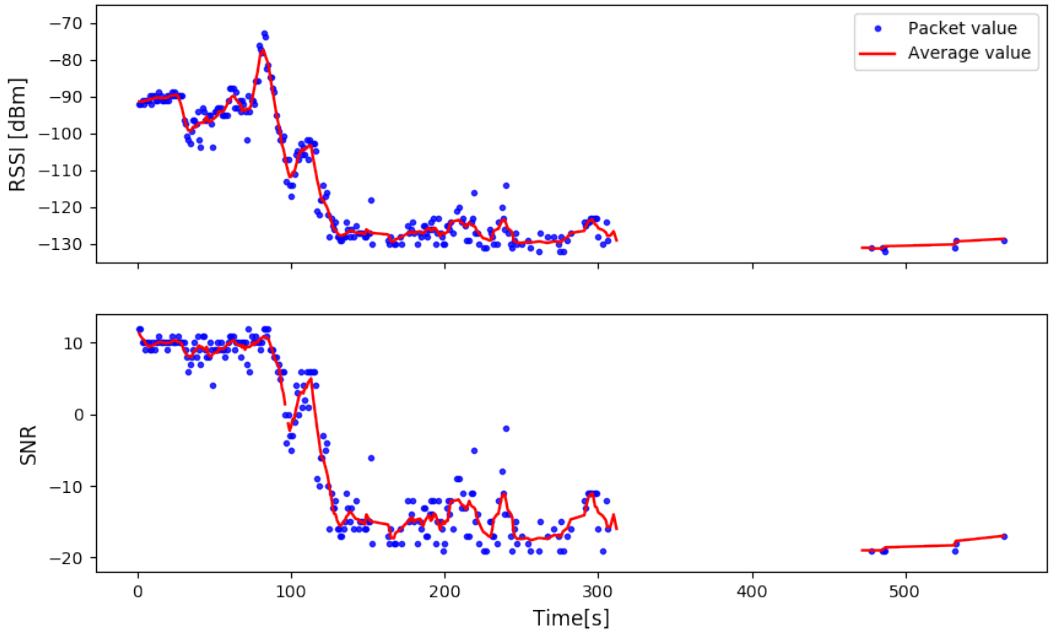

Figure 6.4: The RSSI and SNR values of the first preliminary range test.

As can be seen from plot 6.4, the signal is in the beginning very strong, but then rapidly drops and maintains the same strength during almost the whole of the remaining measurement process. This rapid drop happened after the right turn of the moving mobile node, denoted by the red oval in figure 6.2. From this point on, the signal path gets obstructed by more buildings, thus

the signal strength drops. The complete loss of the signal occurred around the area which is colored red and orange on the path of the moving node. At this area a slight depression in the ground occurs, before the road starts rising again and the signal gets strong enough to be detected, but the packet loss is quite big. From this point on, a few packets were received, before we reached the "End point", a little more than one kilometer away, where the signal was completely lost.

6.1.2 Test in Italy

After a few upgrades of our research platform and the implementation of the Android GPS position logging application, we decided to do second test. This time the transmitter was placed on mount Matajur, on the window shelf of the mountain hut, 1320 meters above the sea level and with a nice line of sight in the direction of the valley and the sea.

Because our goal was to achieve the maximum signal range, we decided to use the 31.25 kHz bandwidth (the minimum bandwidth value that we could use with our hardware), spreading factor 12, coding rate 4/8 and CRC turned on as the LoRa parameter settings for this measurement. With these parameter settings, the packet air-time increased to 3700 milliseconds and the bit rate was only 45.8 bps. We started this test from the parking lot some 40 meters below the hut and drove, with the receiver placed on the car roof, downhill and away from the base of the mountain towards the sea (figure 6.5). The path, which we were following, was almost entirely (besides the first few kilometers) in the transmitters line of sight (figure 6.6). It is also worth noting that the area from the base of the mountain all the way to the end point is completely flat with almost no elevation in between.

At this development stage we still had some trouble with the research platform which did not log the GPS position and measured data for the first few kilometers after we started driving. Unfortunately, we did not notice this problem during the measurement and we continued driving. The part of the path, where the receiver position and measured data were not saved,

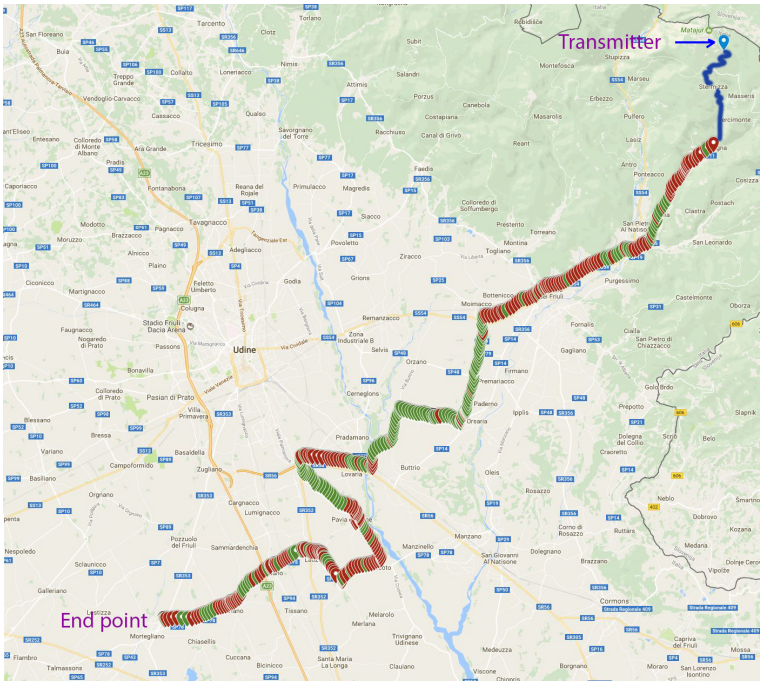

Figure 6.5: The path traveled during the second preliminary test. The green markers represent the locations where the packets were received, the red ones represent locations where the packets were not received, while the blue line represents the path where we did not get measured data due to technical problems.

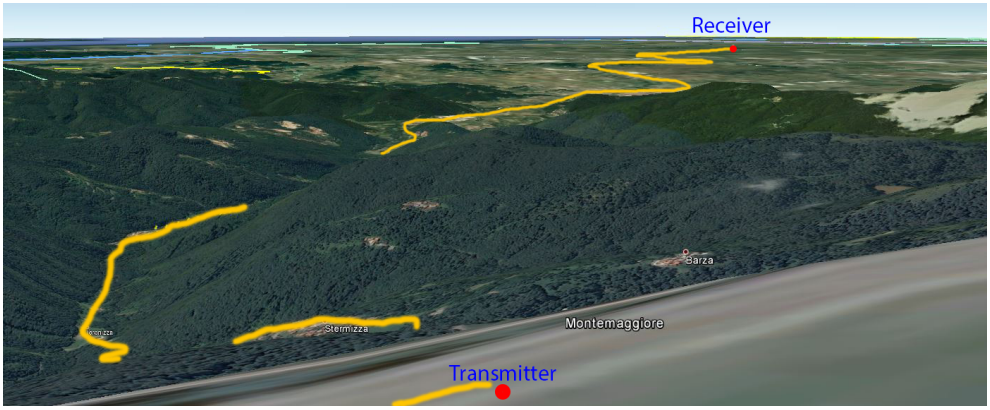

Figure 6.6: A 3D view of the traveled (yellow) path the during second preliminary test. The red dots represents the placement of the transmitter (a bottom one) and the final point of receiver (a top one).

is marked with the blue line on figure 6.5 - from the transmitter to the first red pin. This part is almost entirely in the shadow of the hill and there was probably no LoRa signal reception.

The signal in the valley was quite strong, but there were a lot of spots, where it was completely obstructed by the surrounding area. As the receiver reached the transmitters line of sight, the packet reception completely changed. There was a strong signal and almost no packet loss (see the plots on figure 6.7 between the 40th and 70th minute of the measurement). On the last quarter of the path, the signal becomes weaker and the packet reception drops. This part of the path has a bigger density of villages and towns which contributed to a greater signal loss.

After a little more than one and a half hours into the test, we encountered another technical problem. During this measurement, the signal strength and packet reception was monitored using a laptop. When we reached the point marked "End point" on figure 6.5, laptop's battery died. As we could not see the signal strength and packet reception, we decided to terminate the measurement.

Despite the bad luck, we drove quite far and at this point the signal was

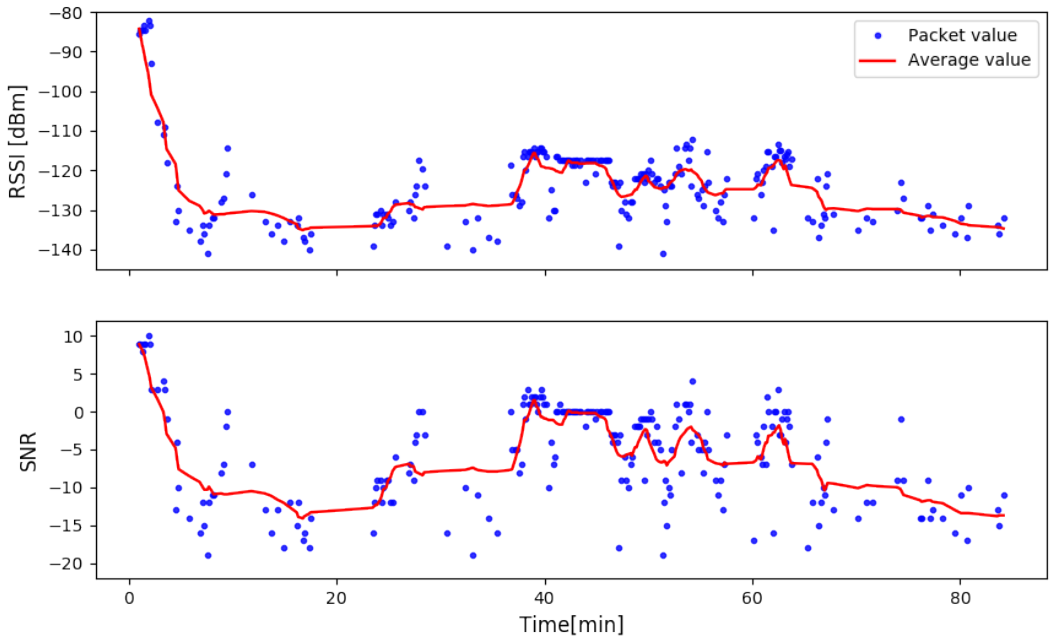

Figure 6.7: The RSSI and SNR values of the second preliminary range test.

still strong and the packet reception rate was satisfying. The air distance from this point to the transmitter is the 39 kilometers. This was quite a surprise, because there had been no papers published up till that time with that kind of results achieved on land. The closest to this had come the authors of the paper [20] with measured distance of 15 kilometers.

This test showed us that the LoRa signal has a great range as long as there is a line of sight between the transmitter and the receiver. If there is a slightly bigger obstacle on the signal path (a house for example), the signal is to weak to penetrate it.

6.2 Final measurements

In this final version of our research platform, we introduced the modem (packet) status logging (for details see section 5.1.2). This helped us to better understand what was happening with packets on the receiver side

during measurements. With that feature we can also provide the answers to the questions like:

- Was the packet transmitted and not detected?
- Was it transmitted and detected, but for some reason not received?
- Or was it transmitted and received?

When the research platform was finished and extensively tested we concentrated on the indoor and urban measurements.

6.2.1 "Grid" measurements behind the building

It is important to know how a signal propagates through and behind the buildings, when one wants to place sensors in an urban area. With that in mind we conducted two measurements with the same LoRa parameter settings and a different placement of the transmitter.

Figure 6.8: The measurement A: The placement and the reception rate of each node when the transmitter was positioned on the far side of the adjacent building. The point numbers are ordered (descending) by the reception rate.

For these measurements, we used three devices: the transmitter (placed on the sixth floor of the building), the control (stationary) receiver and the

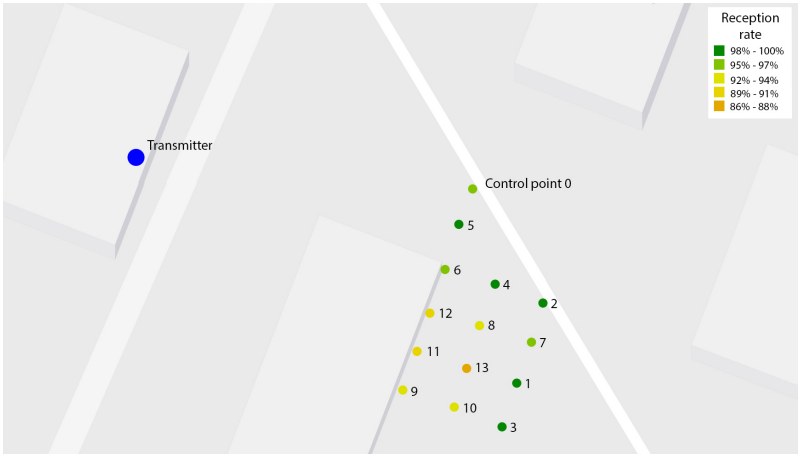

Figure 6.9: The measurement B: The placement of the nodes and the reception rate of each node, when the transmitter was positioned closer to the receivers. The point numbers are ordered (descending) by the reception rate.

receiver that we moved from point to point (let's call it a "moving" receiver). For the first measurement (measurement A), the transmitter was positioned on the far side of the building (relative to the receivers) as can be seen from figure 6.8, while for the second one (measurement B), the transmitter was placed on the opposite side of the building (figure 6.9), closer to the receivers. It is worth noting, that this is reinforced concrete building with two walls in between these two transmitter locations. Our two receivers were placed on the strategic places around one corner and behind the adjacent three-story building (some 40 meters from the building where the transmitter was positioned) as figures 6.8 and 6.9 show. The numbered dots (1 to 13) in these figures represent all the spots where the "moving" receiver was placed and the dot labeled "Control point 0" is the position of the control receiver. In this case, having had a control receiver helped us understand whether the packet reception rate on some point dropped because the signal was blocked by the building (if the packet was received on the control receiver and not on the "moving" receiver) or because some external interference (if the packet was not received on both receivers).

During the measurements, the "moving" receiver was re positioned every 5 minutes from one spot to another, which was roughly 5 meters away. For both tests, the LoRa parameters were set to 250 kHz bandwidth, spreading factor 6 and CRC on with the coding rate of 4/6 (the resulting packet air-time was 9.8 milliseconds with the bit rate of 15.62 kbps). These settings were chosen because the signal was just weak enough, so it did not reach behind the building when the transmitter was positioned for the measurement A. For these tests we also logged the packet statuses which told us is the packet was not detected, if the packet was detected but not decoded for some reason or if the packet was detected and decoded (successfully received). For a detailed explanation what different statuses mean, please read section 5.1.2.

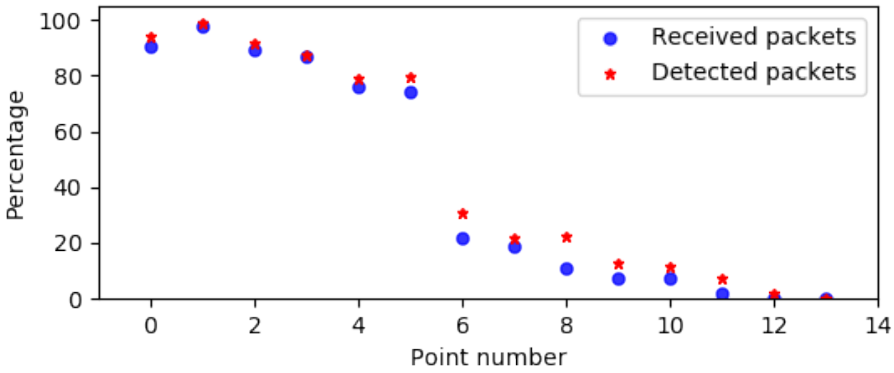

Figure 6.10: The received (detected + decoded) vs the detected packet percentage for the measurement A.

By comparing of the results of measurements A and B, we can see that a very different packet reception was achieved by just placing the transmitter on the opposite side of the building. When the transmitter was placed on the far side of the adjacent building, there was no packet reception on the spots numbered 12 and 13 (see figures 6.8 and 6.10) and a very low reception rate (below 30%) was achieved on the spots numbered 7 to 11. While on the other hand, with the transmitter placed closer to the receivers, the reception rate did not drop below 80%, even behind the building (figures 6.9 and 6.11).

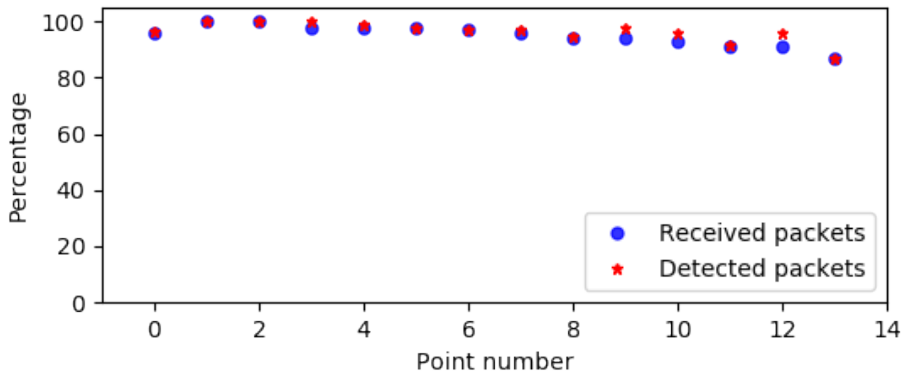

Figure 6.11: The received (detected + decoded) vs the detected packet percentage for the measurement B.

In this measurement, the lowest reception rate was on the spot number 13. The reason for this seems to be the external interference, because at that time, the reception on the control receiver also dropped.

Despite the fact that the measured RSSI and SNR values on the control receiver were very constant for both measurements (figure 6.12), the packet reception rate did not reach 100%, even in this position where there were no obstacles between the transmitter and the receiver. The overall packet reception rate on this receiver during the measurement A was 90.7% and 96.1% for the measurement B.

We can see from the figure 6.13 that in the case of the measurement A the majority of the packets that were not received, were not detected either. This can be, probably, mainly attributed to the position of the transmitter (on the far side of the building) while the surrounding interference is not excluded either. On the other side, there is only a small window (50 packets which is 150 seconds) during the measurement B, where no packet was received, which was probably due to the external interference.

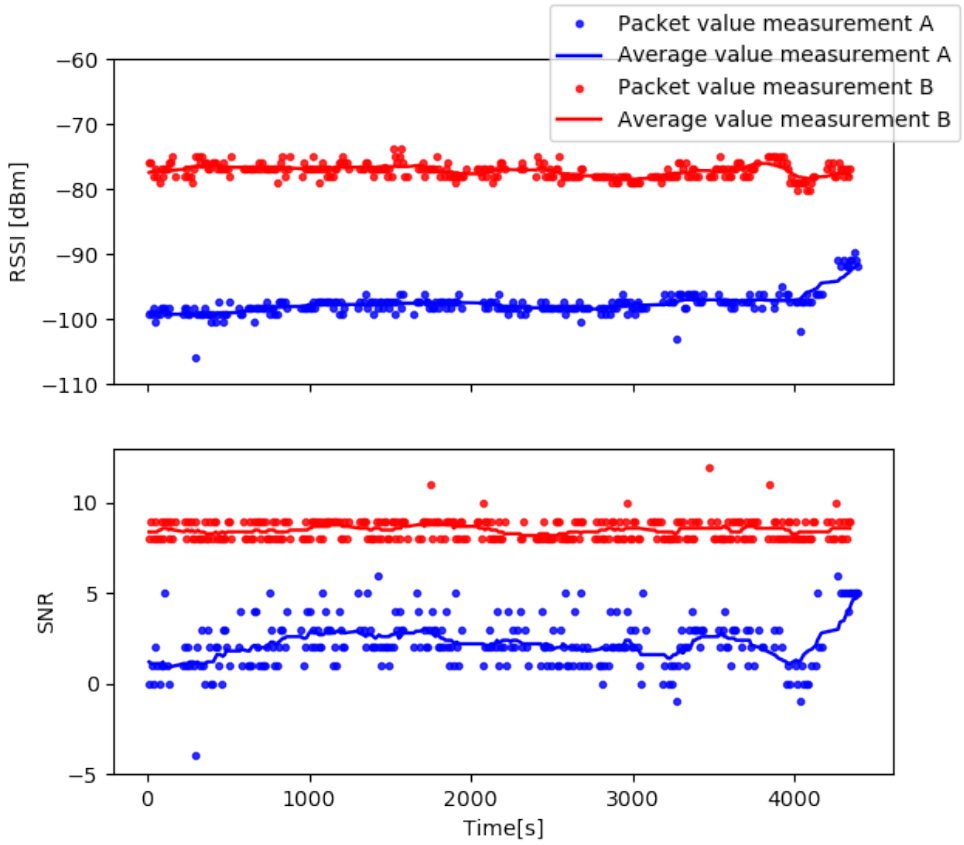

Figure 6.12: The measurement of A's and B's RSSI and SNR values measured on the control receiver.

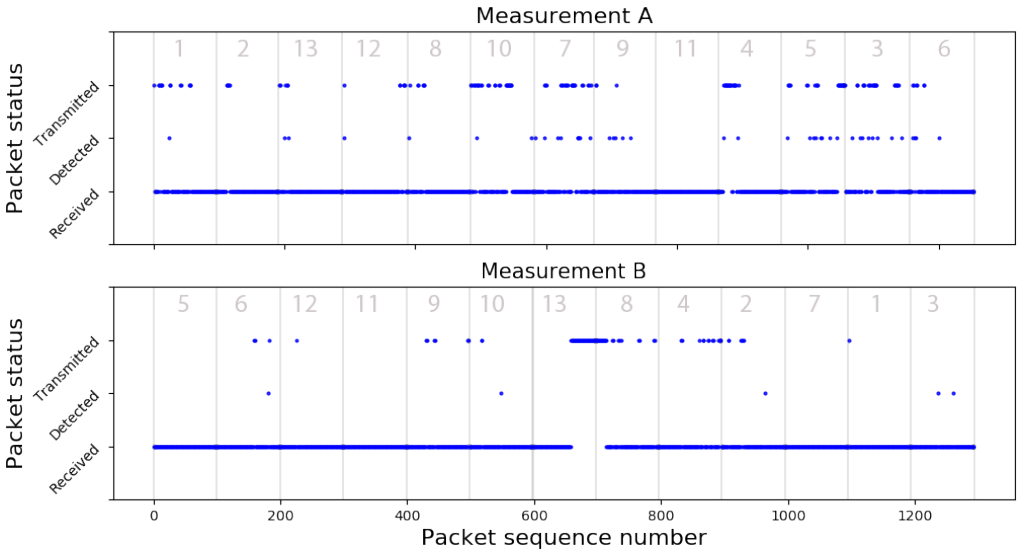

Figure 6.13: The packet statuses for the control node (both measurements). The small window of the unreceived packets (between 650th and 750th sequential packet) is probably the cause of the temporal external interference. The numbered vertical stripes on each plot represent position the number where "moving" receiver was at that time. The Y-axes statuses meaning: *Transmitted:* the packet was only transmitted and not detected or received, *Detected:* packet was transmitted and detected, but not received, *Received:* the packet was transmitted and decoded with the correct CRC.

6.2.2 Indoor measurements at the Faculty of Computer and Information Science

For the next test, we decided to do indoor measurements in the building of the Faculty of Computer and Information Science in Ljubljana. This building is three-story high with the open space in the middle which spans through the entire length, from 1st floor all the way to the roof. On the side of this open space are the hallways, where these measurements were conducted.

For the better understanding of the signal propagation, we conducted two measurements with different LoRa parameter settings. For the first measurement (the measurement C), we chose to use the same settings as we did for the "grid" measurements (250 kHz bandwidth, spreading factor 6 and CRC on with coding rate 4/6 - this gave us the packet air-time of 9.8 milliseconds and the bit rate of 15.62 kbps), while for the second one (measurement D), we chose 125 kHz bandwidth, spreading factor 10 and CRC on with coding rate 4/6 (the resulting packet air-time with these settings is 264 milliseconds and the bit rate of 813 bps).

In these measurements we also used, for the first time, bi-directional communication between the stationary and the moving transceiver (for a detailed explanation of how this mode works see section 5.1.3). The first one was positioned on the third floor, on the south side of the building, while the second one was moving through hallway on each floor, from the south to the north side. In bi-directional mode each transceiver (device) acts as the transmitter and receiver at the "same" time (the device transmits the packet and then listens for the next packet for N second and then repeats the cycle). This is useful for simultaneous testing of uplink and downlink communication between devices involved, without the need for conducting the communication measurement in each direction by it self. For these measurements, the control receiver was placed on the third floor, three meters apart from the stationary transceiver. Because the bi-directional mode was used, we will discuss the results of each measurement in two parts, as if we had measured the uplink and downlink communication separately:

- part one will cover the case where the transmitter was stationary and the receiver was moving (this is equal to downlink),
- while in part two, the case where the transmitter was moving and the receiver was stationary will be discussed (this is equal to uplink).

First measurement

We will first discuss the results of the measurement C where we have fixed the transmitter and the receiver which moves through the hallways. We will denote this measurement the "measurement C1" for easier referencing.

As said before, we placed the transmitter on the third floor, on the south side of the building, at the height of two meters, while the control receiver was on the same height, three meters away, on the opposite side of the corridor (figure 6.14).

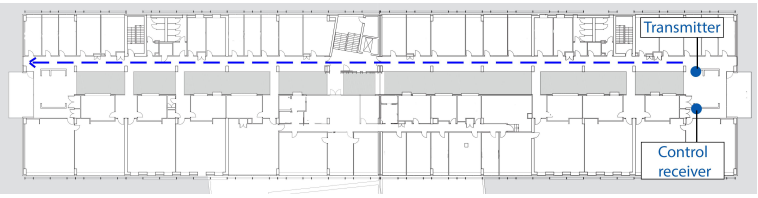

Figure 6.14: The placement of the transmitter and the control receiver. The blue dashed line is the path of the moving receiver.

With these two devices in place, we started the transmission and slowly walked, with another receiver down the hallway of the third floor, away from the transmitter. When we reached the end of hallway, we moved to the south side of the second floor (beneath the transmitter) and repeated the measurement. We did the same on the first floor.

The results were almost as expected. There was a pretty good packet reception rate through the whole building (figures 6.15), which did not drop below 70%. By comparing both plots in figure 6.16, we can see that most of the packets were missed by only the moving receiver. This is probably the

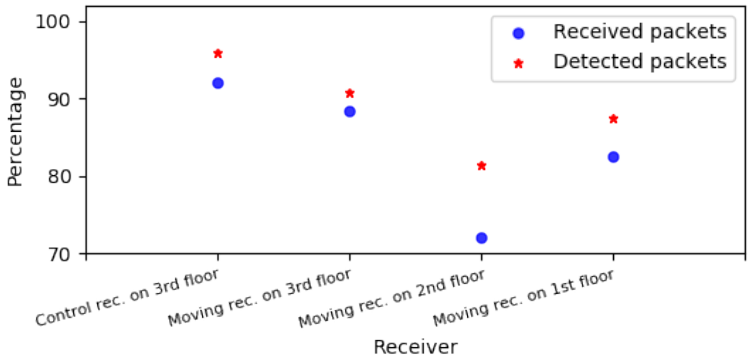

Figure 6.15: The comparison of the reception rate for each floor during the measurement C1.

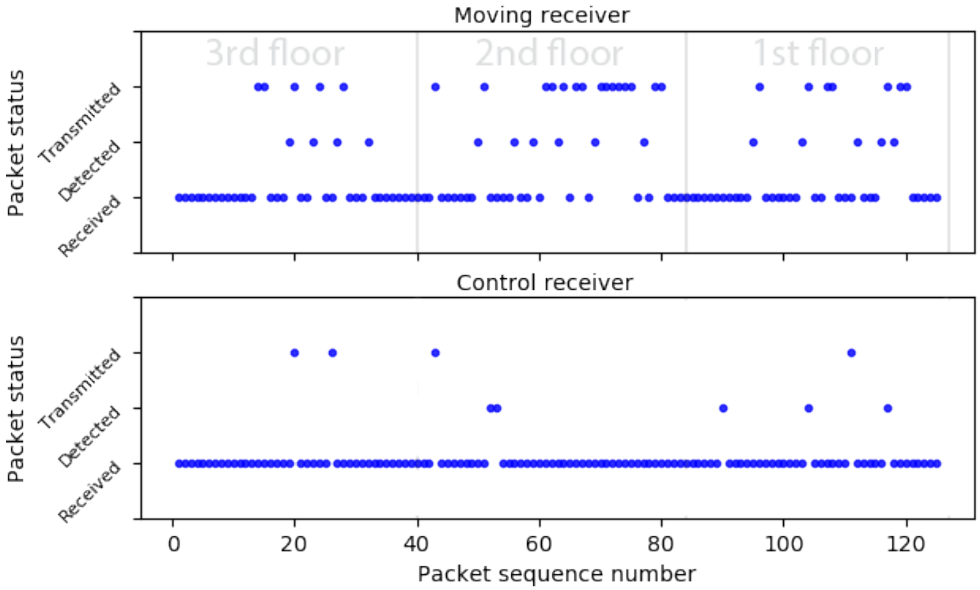

Figure 6.16: The transmitted, detected and received packets for all of the floors as measured on the moving receiver (top) and control receiver (bottom) during the measurement C1. Each section with a floor number represents the position of the moving receiver at that moment.

consequence of different propagation paths or some temporal interference in the vicinity of the moving receiver and not in the vicinity of the transmitter (or the control receiver).The fact that the RSSI values are slowly decreasing in respect to the distance (figure 6.17)is also not surprsinig. More unusual are the SNR values which stay, during the first half of the measurement, for all of the floors almost the same and then suddenly drop. Again, this is probably the consequence of the surrounding interference, because the hallway surrounding area in the second half of the building is almost identical to the first half. More surprising is the fact, that if we compare the results between the floors, these values do not change as much as they do with the distance. This is probably the result of the partially opened hallways on the third and on the second floor, and completely opened hallway on the first floor and the fact that the building is a 100 meters long.

Because in this measurement the bi-directional mode was used, we also measured the signal propagation in the direction from the moving node (in this case the transmitter) to the stationary node (the receiver). In other words, we measured the uplink communication. Lets call this part of the measurements the "measurement C2". Because we only used three devices and we placed two devices on the third floor, we did not have another one close to the moving transmitter, hence we did not have the control receiver for this part of the measurement but only two receivers. We will denote the receiver (device) that was in the measurement C1 the transmitter as the 1st stationary receiver and the one that was control receiver as the 2nd stationary receiver.

In this measurement, the RSSI values are slightly higher for the whole time, while the SNR values are the same in first half of the hallways and higher in the second half (figure 6.18). They all follow the same pattern: the signal gets weaker in respect to the increasing distance, but its strength changes only a little on different floors. The packet reception rate is this time, in contrary to the RSSI and SNR values, significantly higher on all of the floors, especial on the second one (figures 6.19 and 6.20). We do not have a

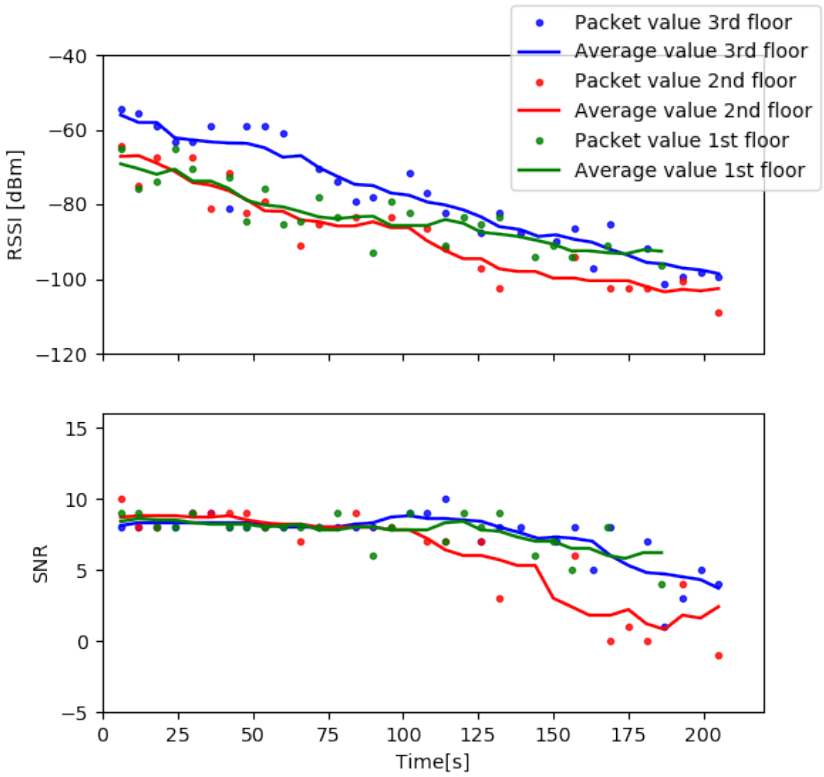

Figure 6.17: The RSSI and SNR values of the measurement C1 for all of the floors.

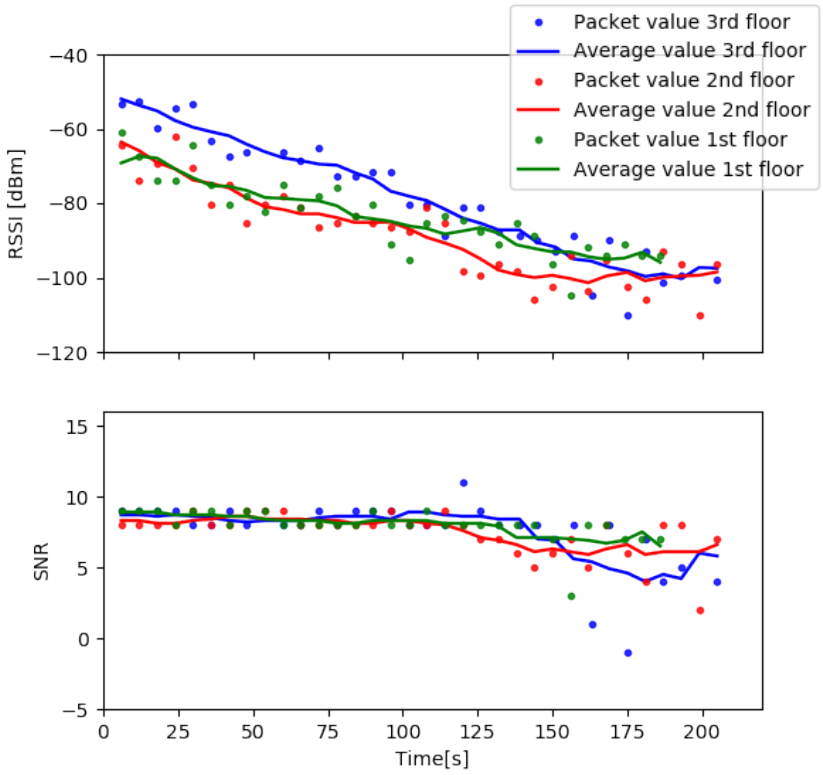

Figure 6.18: The RSSI and SNR values of the measurement C2 (moving transmitter) for all of the floors.

good explanation why there is such a difference between the measurements, since both were conducted almost simultaneously. More research on this topic would need to be done.

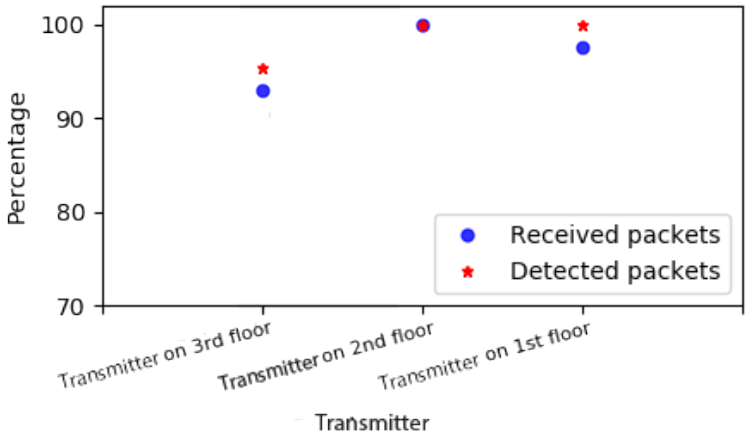

Figure 6.19: The comparison of the reception rate on 1st receiver when the mobile transmitter was positioned on a different floor during the measurement C2.

The following figures numbered 6.21 through 6.26 present the results of the measurements C1 and C2 on a building blueprint for an easier interpretation. The measurements were conducted (as shown in figure 6.14) from the right to the left side of the building. On each individual figure the results of both measurements C1 (the top row of colored dots) and C2 (the bottom row of the colored dots) are presented. These figures nicely show how the signal strength is decreasing with the respect to the distance. The color scale on the right side of the figures represents the RSSI or SNR values of the received packets, while the gray dots inthe measurement results represent places in the building where packets were not received.

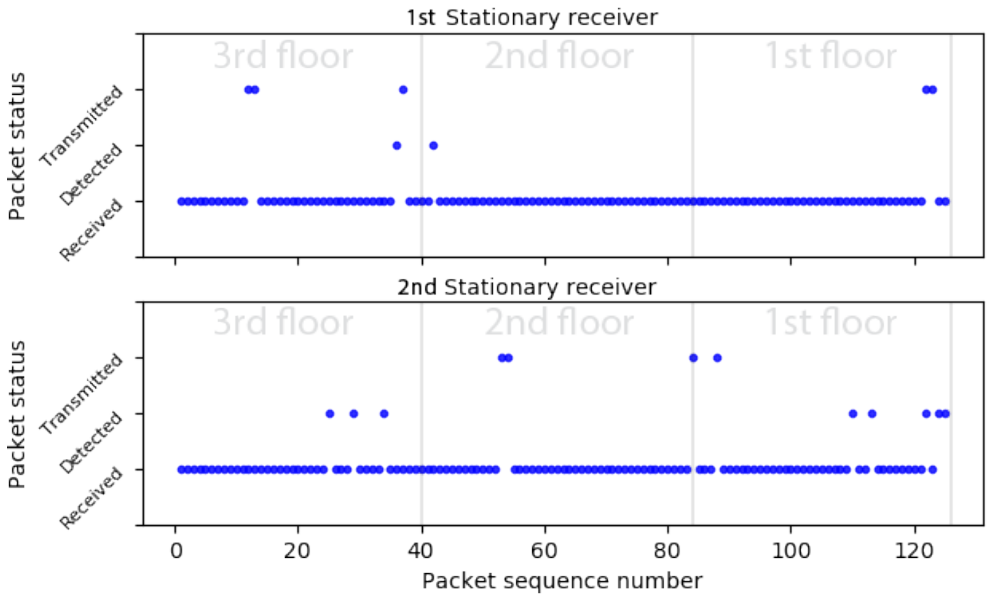

Figure 6.20: The transmitted, detected and received packets for all of the floors as measured on the 1st stationary receiver (top) and the 2nd stationary receiver (bottom) during the measurement C2. Each section with a floor number represents the position of the moving transmitter at that moment.

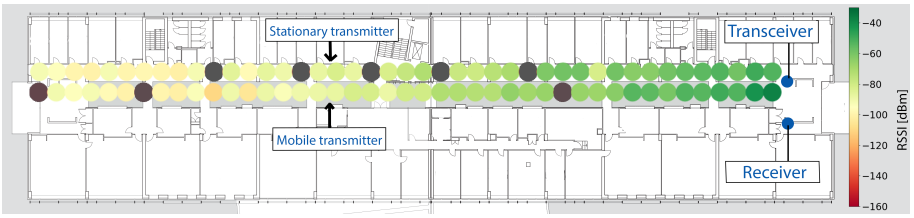

Figure 6.21: The RSSI values of both measurements (C1 and C2) for the third floor. The device on this figure denoted by the "Receiver" was in the case of the measurement C1 control receiver and in the case of the measurement C2 the second receiver.

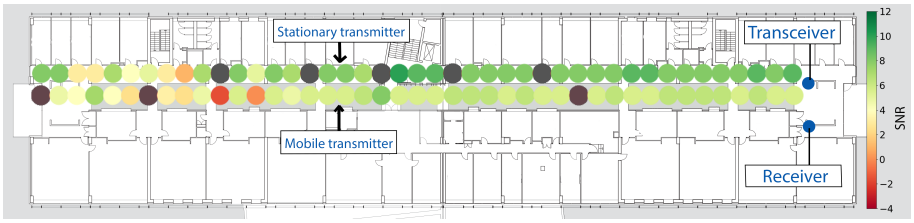

Figure 6.22: The SNR values of both measurements (C1 and C2) for the third floor. The device on this figure denoted by the "Receiver" was in the case of the measurement C1 the control receiver and in the case of the measurement C2 the second receiver.

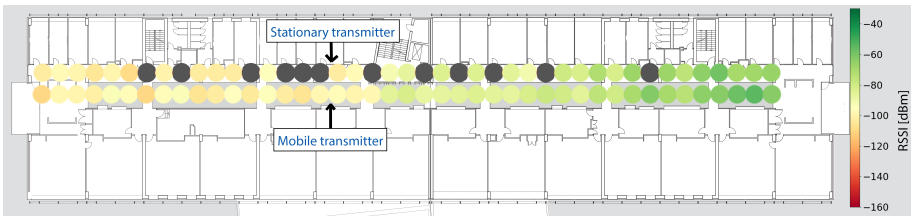

Figure 6.23: The RSSI values of both measurements (C1 and C2) for the second floor.

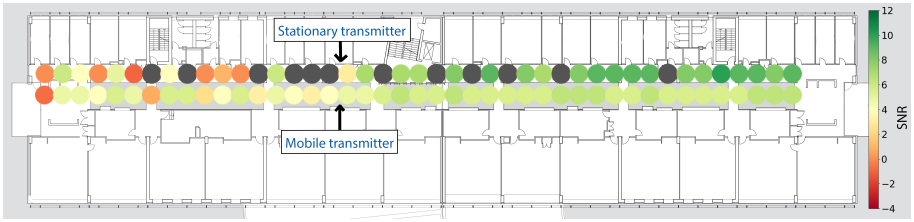

Figure 6.24: The SNR values of both measurements (C1 and C2) for the second floor.

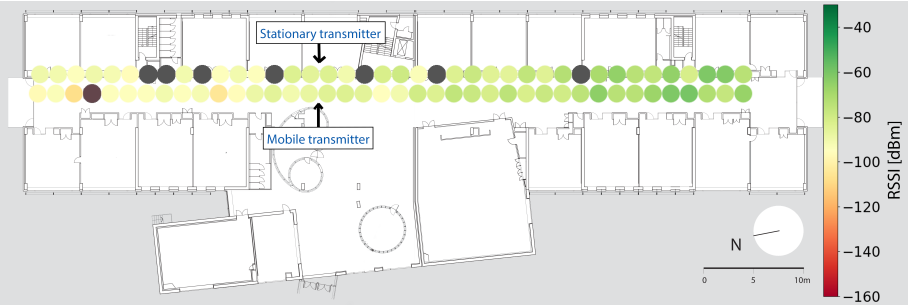

Figure 6.25: The RSSI values of both measurements (C1 and C2) for the first floor.

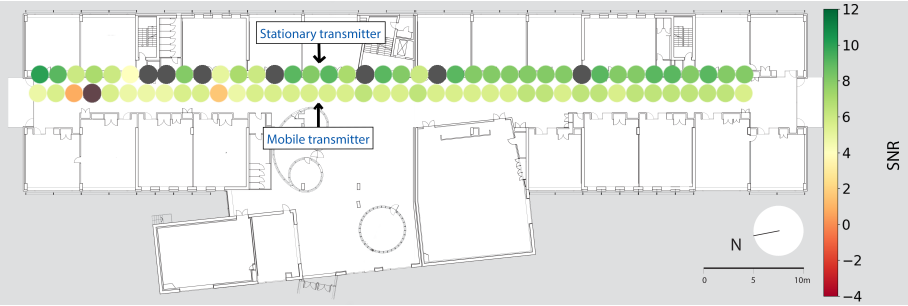

Figure 6.26: The SNR values of both measurements (C1 and C2) for the first floor.

Second measurement

In this section, results of the measurement D will be presented. The path of the moving transceiver, the placement of the fixed transceiver and the placement of the control receiver were the same as they were in the measurements C, all that was changed were the LoRa parameter settings.

As in the previous section, we will begin with an analysis of the results where the transmitter was on a fixed position and the receiver was moving away from the transmitter. This part of the measurement D will be denoted by the "measurement D1".

We can see from the figures 6.27 and 6.28 that the results have drastically changed by just changing the LoRa parameters. In this case, the packet reception rates are much higher (over 95% on all floors) than the ones in the measurement C.

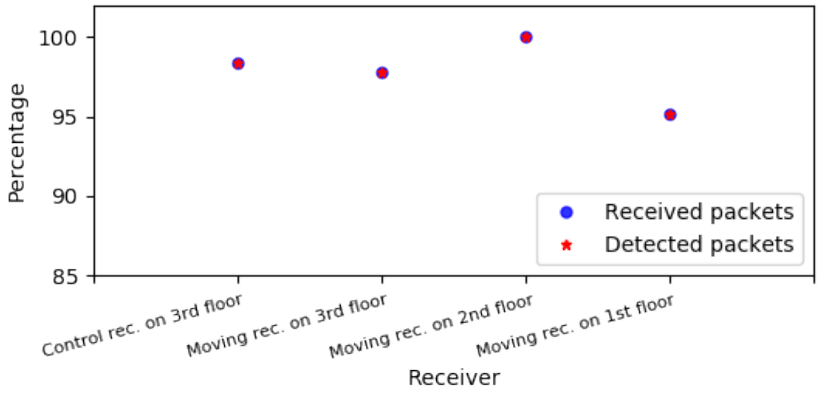

Figure 6.27: The comparison of the reception rate of each floor during the measurement D1.

One would expect, based on the packet reception rate results, that the RSSI values would also be much higher this time, but the data presented in the figure 6.29 show a different picture. The RSSI values of the measurement D1 are almost 10 dBm lower then they were in the measurement C1. On the other hand, the SNR values are higher, but not as constant through all of

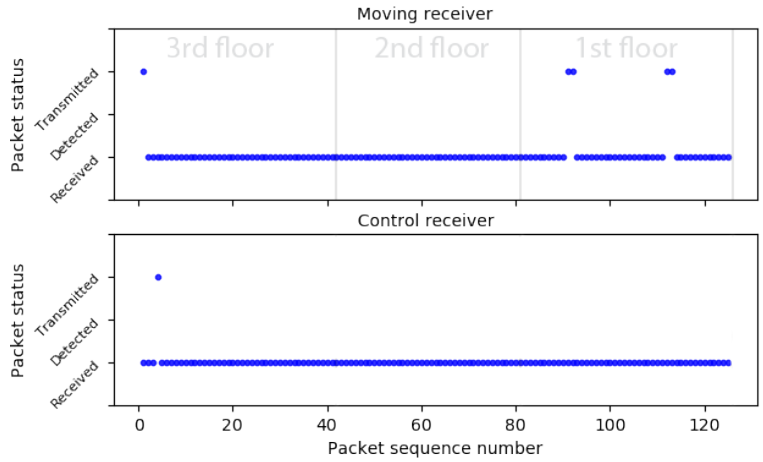

Figure 6.28: The transmitted, detected and received packets for all of the floors as measured on the moving receiver (top) and the control receiver (bottom) during the measurement D1. Each section with a floor number represents the position of moving receiver at that moment.

the floors as thy were in previous measurement. The higher reception rate, despite the lower RSSI values, is the consequence of the very basics of how the LoRa modulation works. With a lower bandwidth and a higher spreading factor the packets get more resilient to the surrounding noise and a higher reception rate can be achieved.

In the measurement D2, we also had (as in the measurement C2) two stationary receivers which will be denoted as they were for the measurement C2. The more constant RSSI and SNR values can be seen in the measurement D2 (figure 6.30), where we have the moving transmitter and the stationary receiver. The RSSI values are decreasing with pretty much the same rate on all of the floors, while the SNR values stay almost the same and do not change as much as they did in the measurement D1. With this settings and moving transmitter, the reception rate has drastically improved. On all of the floors, except on the first one, where the packet reception rate reached 97.7% (figure 6.31), no packet was lost (figure 6.32).

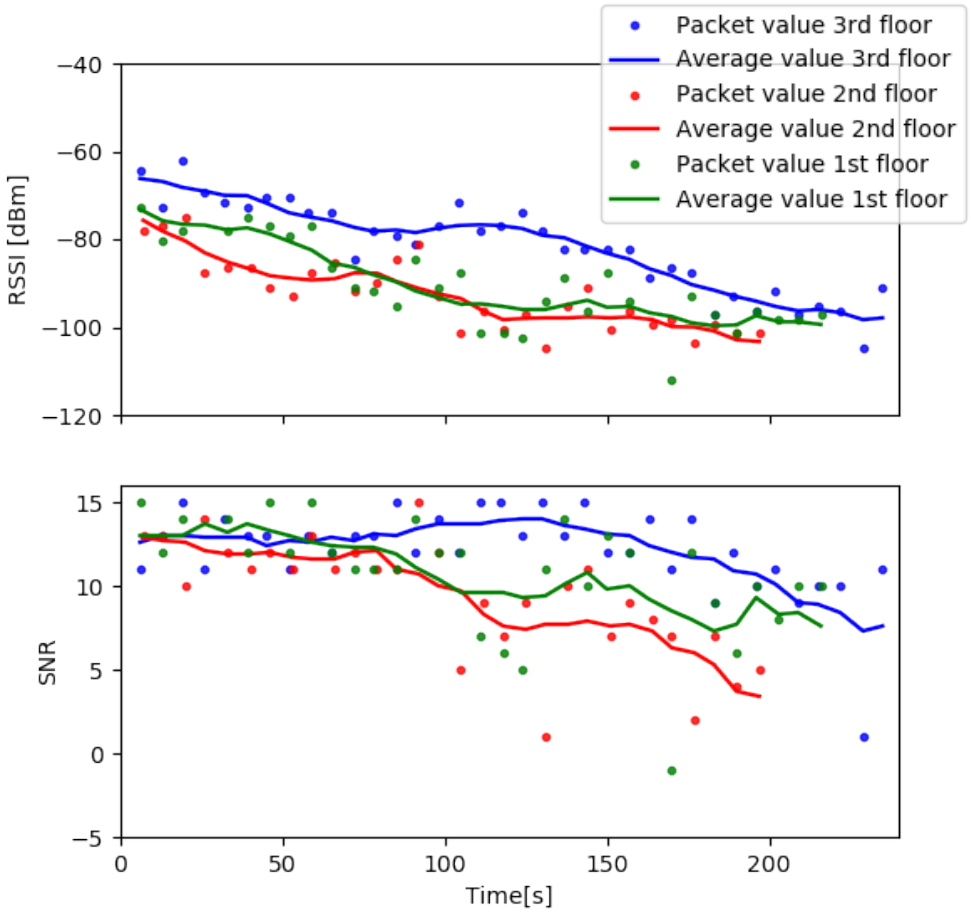

Figure 6.29: The RSSI and SNR values of the measurement D1 for all of the floors.

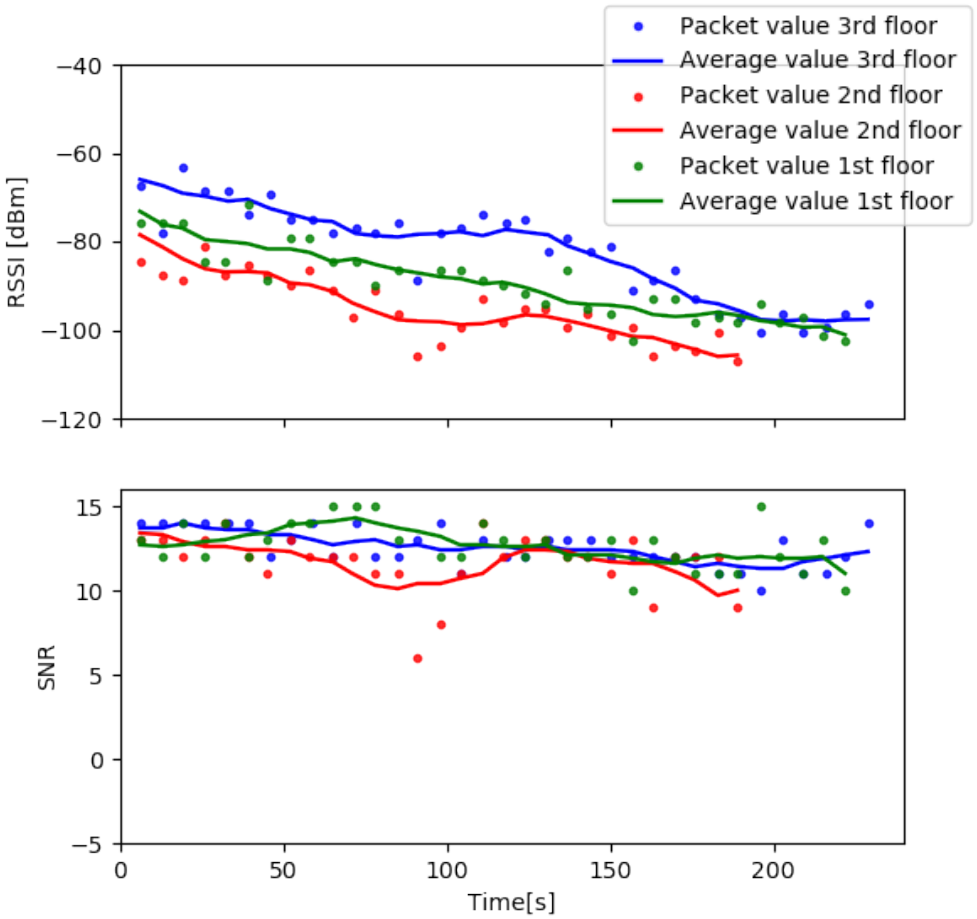

Figure 6.30: The RSSI and SNR values of the measurement D2 for all of the floors.

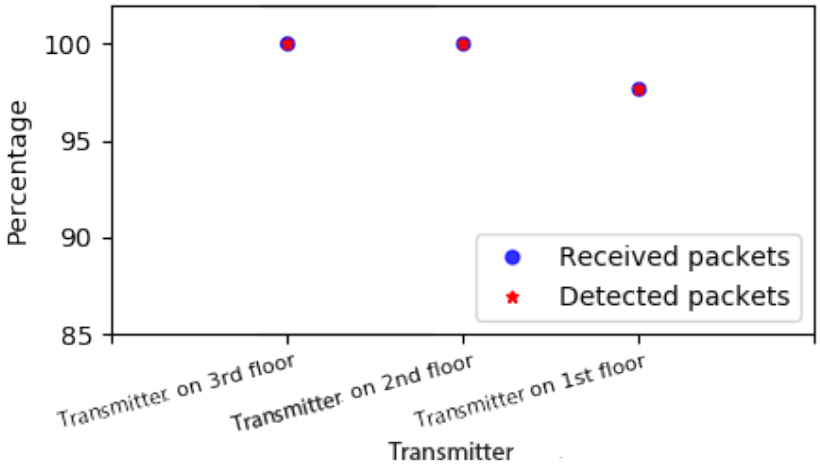

Figure 6.31: The comparison of the reception rate on the 1st receiver when the mobile transmitter was positioned on a different floor during the measurement D2.

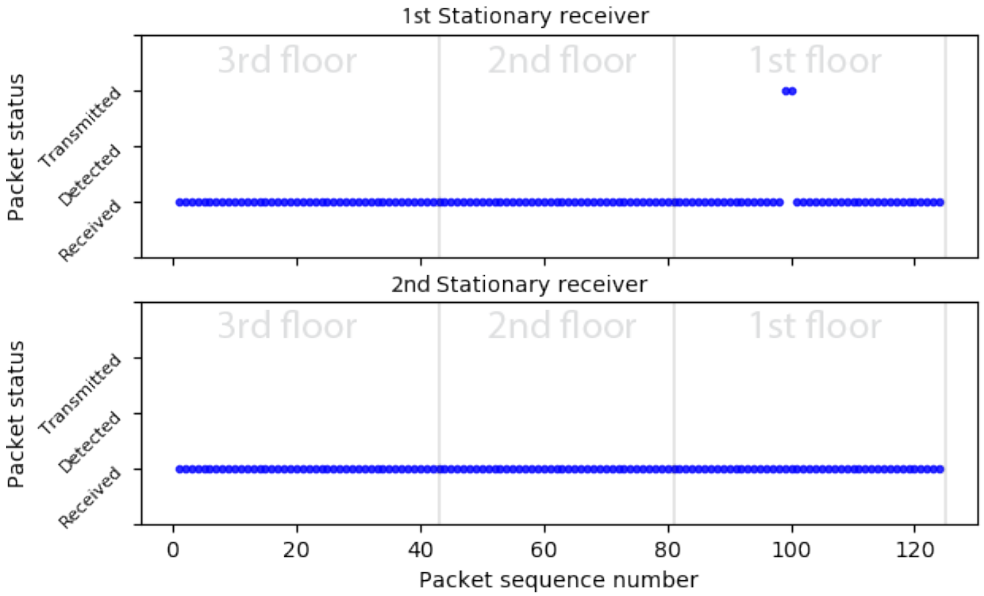

Figure 6.32: The transmitted, detected and received packets for all of the floors as measured on the 1st stationary receiver (top) and the 2nd stationary receiver (bottom) during the measurement D2. Each section with a floor number represents the position of the moving transmitter at that moment.

How good the coverage was and how little the signal varied during the measurement D can also be seen on the signal coverage maps in the figures from 6.33 to 6.38.

Figure 6.33: The RSSI values of both measurements (D1 and D2) for the third floor. The device on this figure denoted by the "Receiver" was in the case of the measurement D1 the control receiver and in the case of the measurement D2 the 2nd receiver.

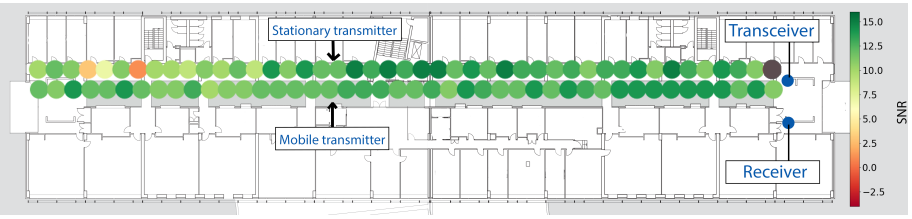

Figure 6.34: The SNR values of both measurements (D1 and D2) for the third floor. The device on this figure denoted by the "Receiver" was in the case of the measurement D1 the control receiver and in the case of the measurement D2 the 2nd receiver.

Figure 6.35: The RSSI values of both measurements (D1 and D2) for the second floor.

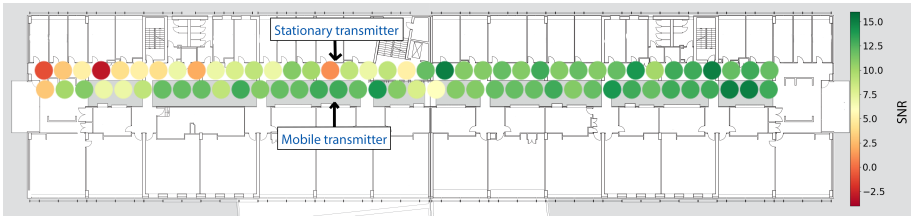

Figure 6.36: The SNR values of both measurements (D1 and D2) for the second floor.

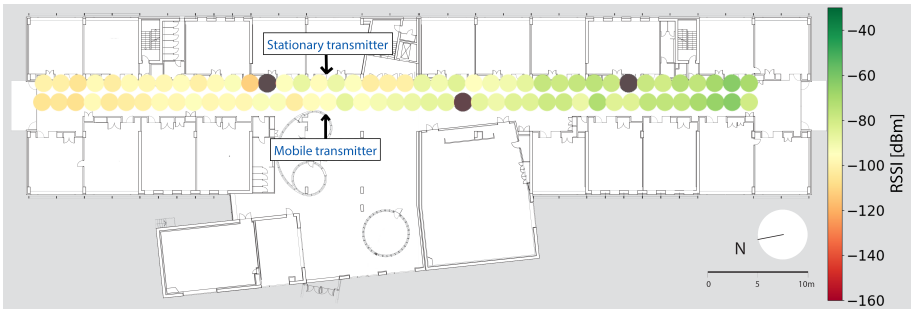

Figure 6.37: The RSSI values of both measurements (D1 and D2) for the first floor.

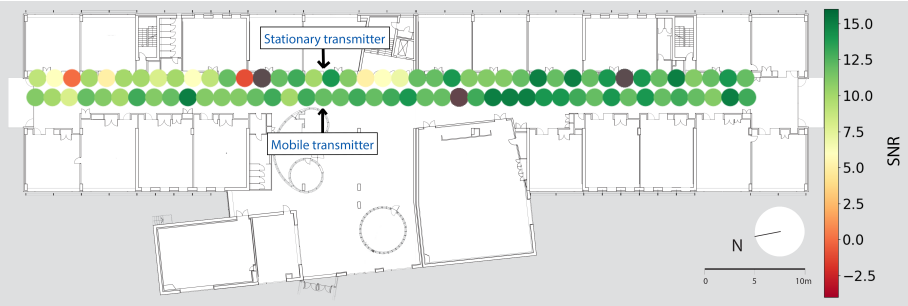

Figure 6.38: The SNR values of both measurements (D1 and D2) for the first floor.

6.2.3 The outdoor-indoor measurements at the Faculty of Computer and Information Science

When we completed the indoor measurements, where all of the devices were inside the building, we also wanted to know what happens if we place one transceiver outside and how does the signal coverage changes in this case.

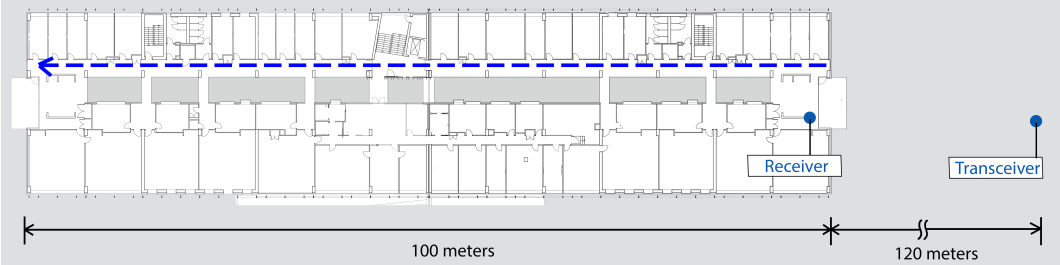

Figure 6.39: The placement of the transceiver, the control receiver (in the case of the measurement E1 or F1) or the second receiver (in case of the measurement E2 or F2) and the path of the moving transceiver. The transceiver positioned outside the building was placed 2 meters above the ground, while the receiver on the third floor of the building was placed near the window at the height of 2 meters.

For this measurement the transceiver was placed 120 meters away from the building, the control receiver was on the third floor near the window

and the second transceiver was moving down the hallways as it was doing in previous measurements. The placement of all of the devices can be seen in figure 6.39.

During this test, the same settings were used as they were in the previous indoor measurements. This time we will denote the first part of the measurements, where we used the LoRa parameter settings with 250 kHz bandwidth, as the "measurement E" and the second one, where we use 125 kHz bandwidth, as the "measurement F". For both measurements the bi-directional communication was also used.

First measurement

Because the bi-directional communication was used during the measurement E, we will first look at the part of the results where we had the stationary transmitter and the moving receiver. This part will be denoted as the "measurement E1".

We can see from the figure 6.40 that this time the signal reception was very poor and the packet reception rate reached a little over 20% only on the first floor. With this set of settings and the placement of the transmitter, the the signal covered only 1/3 of the building (see the plot of the moving receiver in figure 6.41). Such a low reception rate can be caused by the multipath or propagation environment. As we can see from figures 6.40 and 6.41, the reception was not perfect on the control receiver either, despite the fact that it was in the stationary transmitters line of sight.

How bad the reception was, the RSSI and SNR values in the figure 6.42 also show us. In this case, the signal strength changed in the first third of the building as much as it did during the whole measurements C and D, while in the other two thirds it completely disappears.

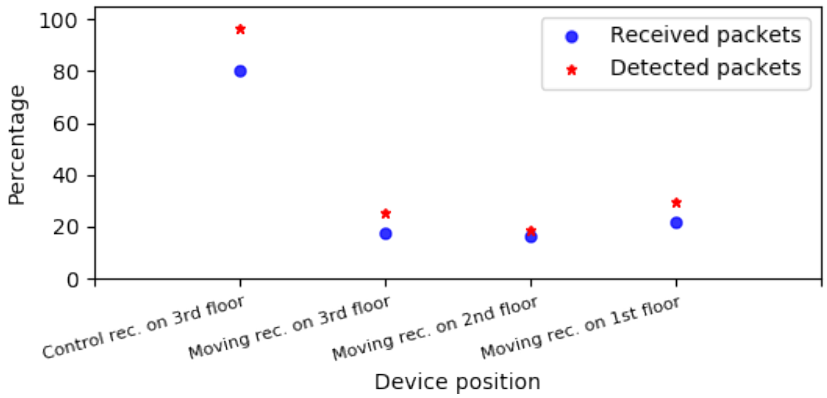

Figure 6.40: The reception rate on each floor during the measurement E1.

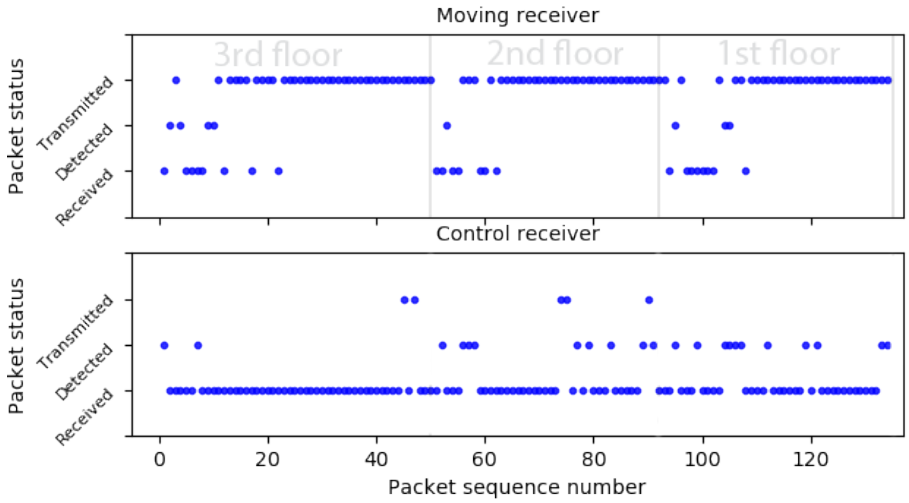

Figure 6.41: The transmitted, detected and received packets for all of the floors as measured on the moving receiver (top) and the control receiver (bottom) during the measurement E1. Each section with a floor number represents the position of the moving receiver at that moment.

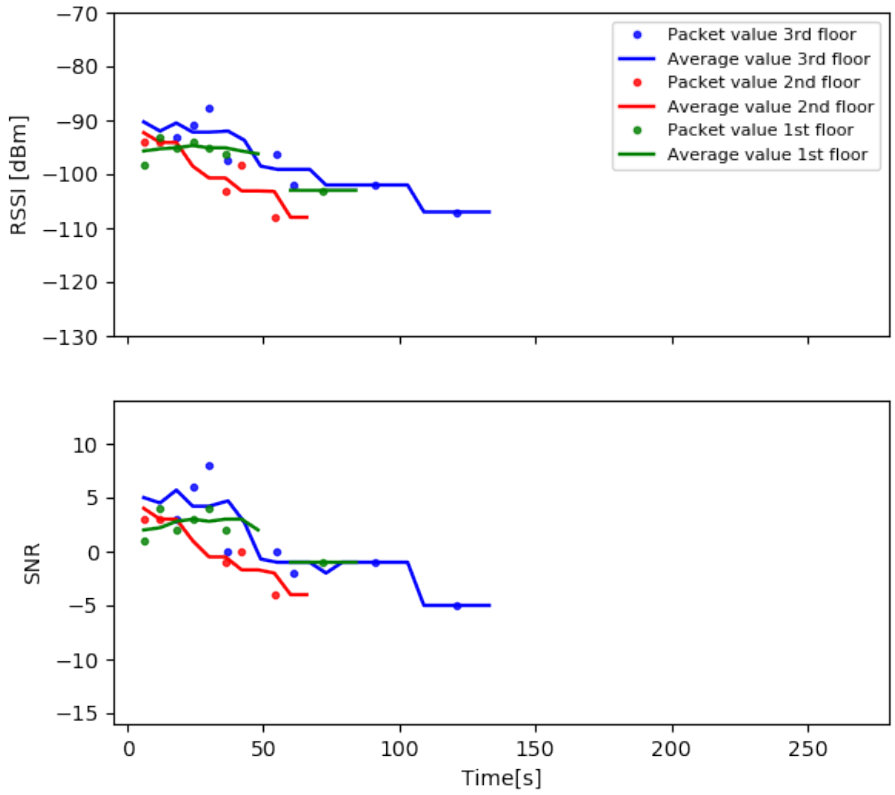

Figure 6.42: The RSSI and SNR values of the measurement E1 for all of the floors. All of the measurements lasted for 260 seconds, but there was no signal reception in two thirds of the building.

The other part of the measurement, where we had the moving transmitter inside the building and two stationary receivers, one on the third floor and one outside the building, will be denoted as the "measurement E2". This measurement represents the uplink communication.

In this case, the signal reception was even worse than it was in the other direction (figures 6.43 and 6.45). When the moving transmitter was on the second floor, the reception rate on the stationary receiver positioned outside, reached only 10%. On the other hand, the reception rate on the 2nd stationary receiver (positioned on the third floor) reached a similar reception rate as it did in the measurement C2 (figure 6.44). Maybe the reason for such a

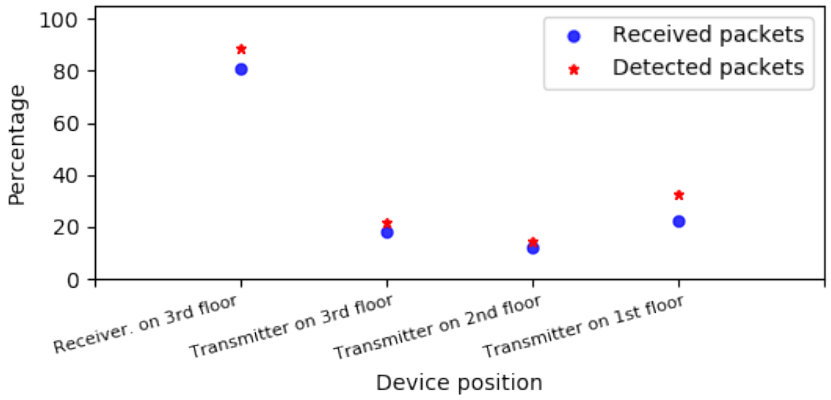

Figure 6.43: The reception rate as measured on the 1st stationary receiver positioned outside and on the 2nd stationary receiver (the "Receiver on 3rd floor") during the measurement E2.

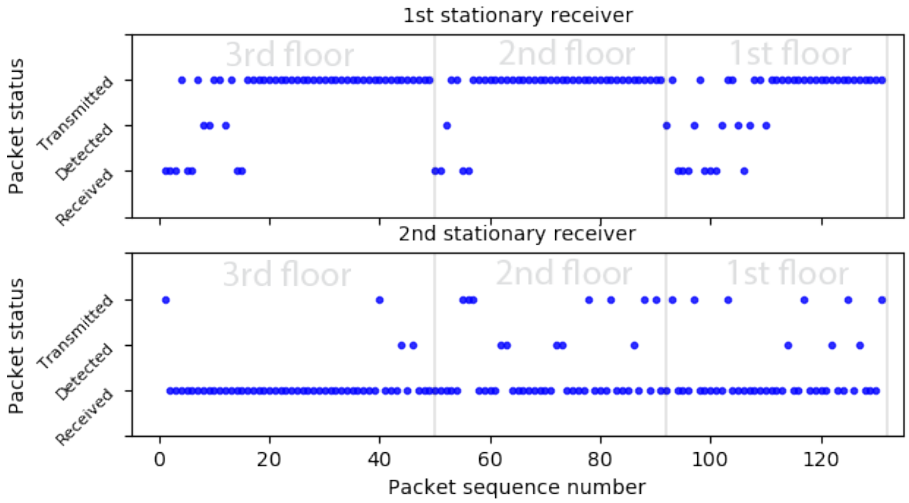

Figure 6.44: The transmitted, detected and received packets for all of the floors as measured on the 1st stationary receiver (top) and the 2nd stationary receiver (bottom) during the measurement E2. Each section with a floor number represents the position of the moving transmitter at that moment.

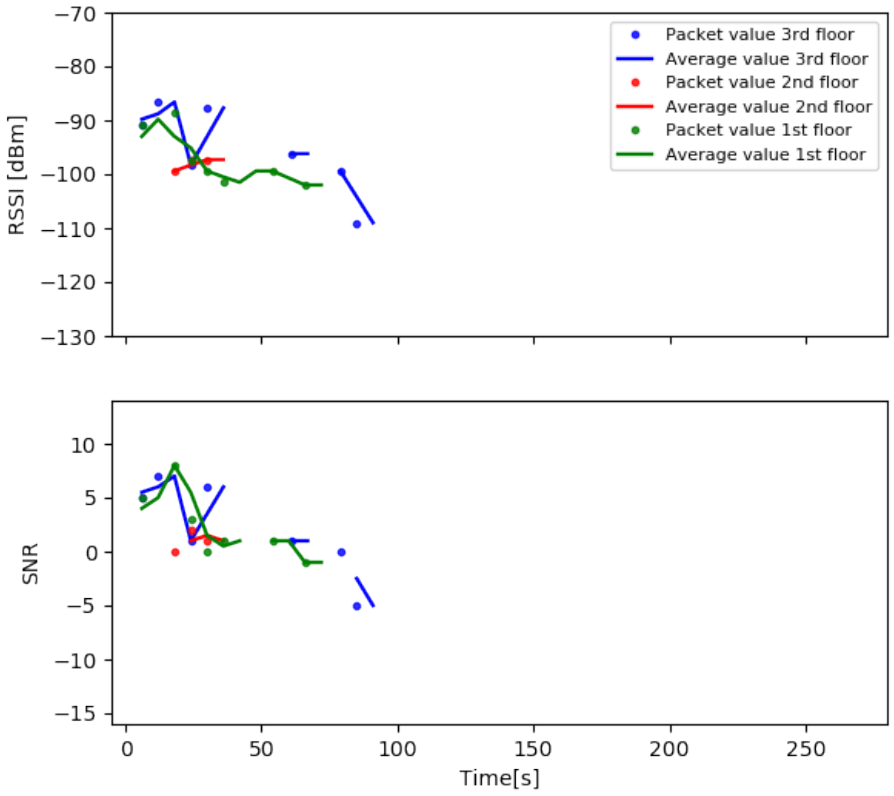

Figure 6.45: The RSSI and SNR values of the measurement E2 for all of the floors measured on the stationary receiver positioned outside. All measurements lasted for 260 seconds, but there was no signal reception in two thirds of the building.

bad reception rate is only a bad positioning of the outside transceiver, which would needed to be tested in the future work.

The signal coverage and its RSSI and SNR values for each floor can be seen in figures 6.46 to 6.51.

Figure 6.46: The RSSI values of both measurements (E1 and E2) for the third floor. The device on this figure denoted by the "Receiver" was in the case of the measurement E1 the control receiver and in the case of the measurement E2 the 2nd receiver.

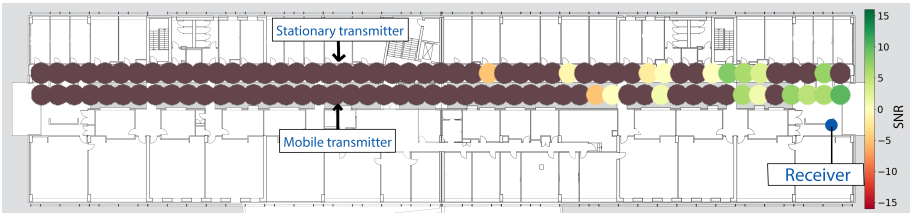

Figure 6.47: The SNR values of both measurements (E1 and E2) for the third floor. The device on this figure denoted by the "Receiver" was in the case of the measurement E1 the control receiver and in the case of the measurement E2 the 2nd receiver.

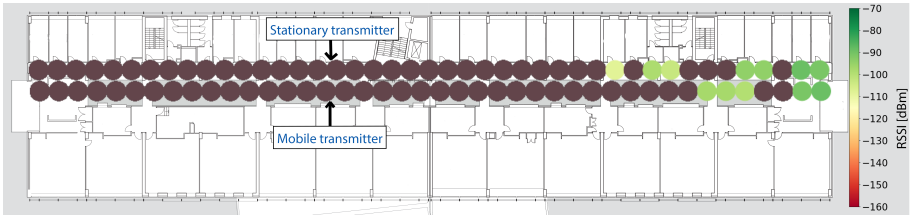

Figure 6.48: The RSSI values of both measurements (E1 and E2) for the second floor.

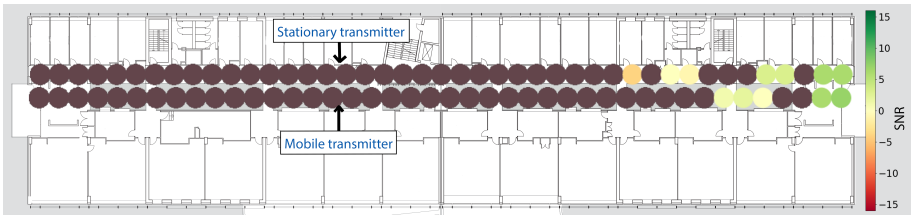

Figure 6.49: The SNR values of both measurements (E1 and E2) for the second floor.

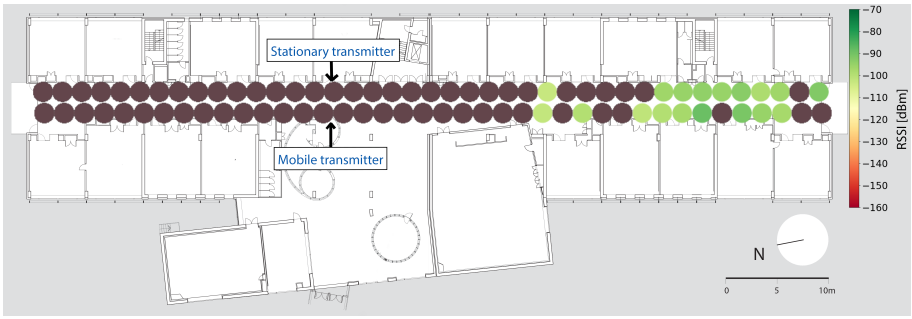

Figure 6.50: The RSSI values of both measurements (E1 and E2) for the first floor.

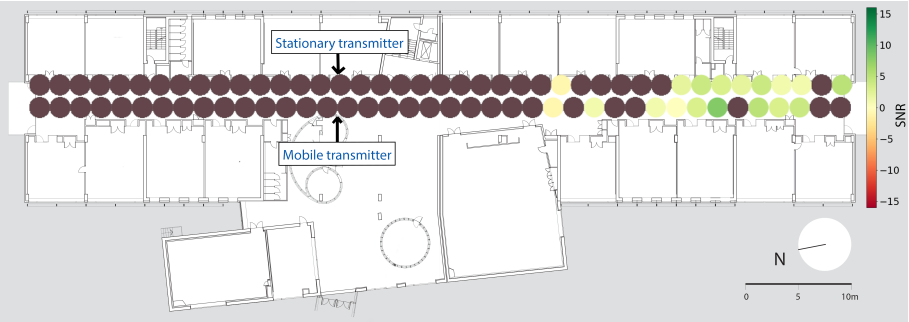

Figure 6.51: The SNR values of both measurements (E1 and E2) for the first floor.

Second measurement

When conducting this measurement (the measurement F), we used the same settings as we did in the measurement D. We will denote the first part of the measurement F as the "measurement F1", where we had the stationary transmitter positioned outside, the control receiver on the third floor and the moving receiver inside the building.

This part of the measurement gave us a very different signal coverage through the building than the measurement E did. The lowest reception rate was this time 50%, while the highest one was a little over 80% (figure 6.52). Such a difference between the reception rates on different floors is the consequence of the height difference between the transmitter and the receiver - the first floor is almost entirely in the transmitters line of sight, while the other two are not and the signal has to travel partially through the concrete floors. Different results can be observed on the control receiver too, where the reception rate reached 100% (figures 6.52 and 6.53).

The RSSI and SNR values are dropping almost linearly in respect to the time (or the distance) (figure 6.54). In this case, the RSSI and SNR values are correlated to each other - the RSSI and SNR plots for each floor have almost the same shape, while in all of the other measurements they do not.

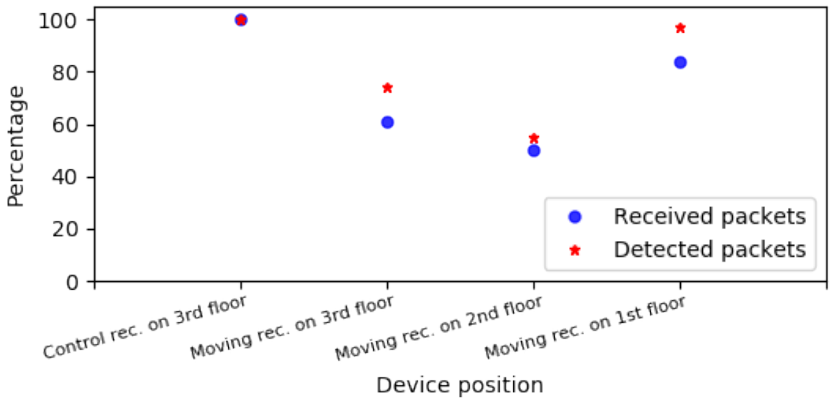

Figure 6.52: The reception rate on each floor during measurement F1.

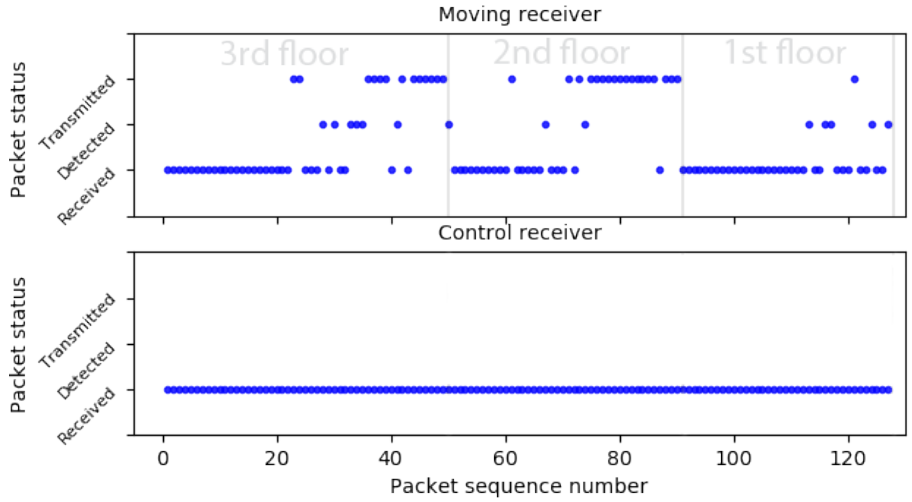

Figure 6.53: The transmitted, detected and received packets for all of the floors as measured on the moving receiver (top) and the control receiver (bottom) during the measurement F1. Each section with a floor number represents the position of the moving receiver at that moment.

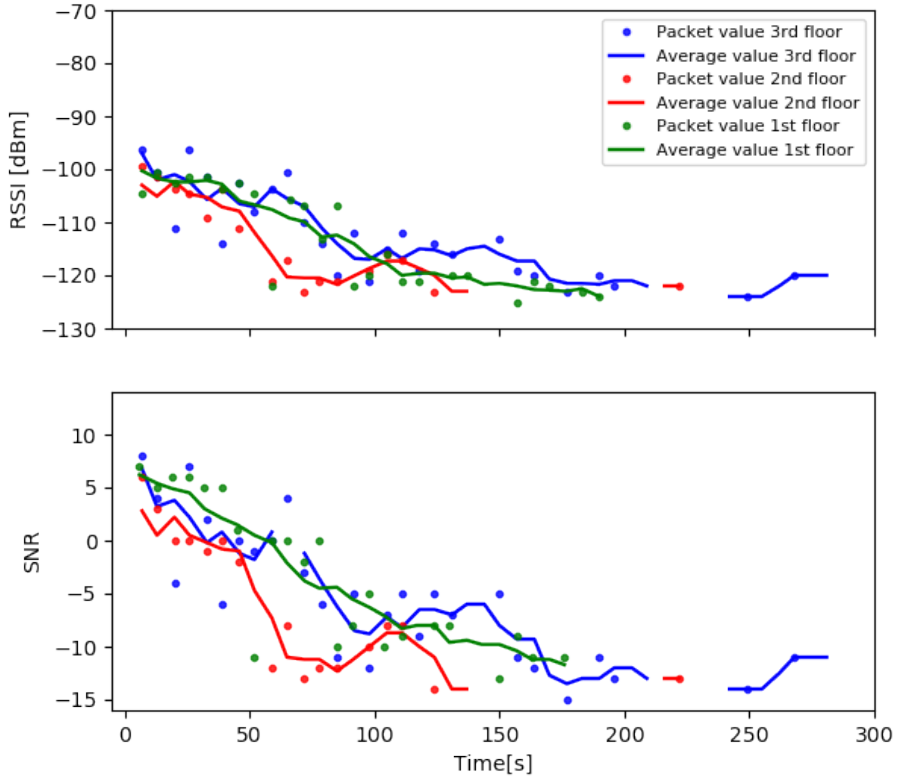

Figure 6.54: The RSSI and SNR values of the measurement F1 for all of the floors. The measurement on the third floor lasted for 260 seconds while the measurements on the other two floors lasted a little less (230 seconds).

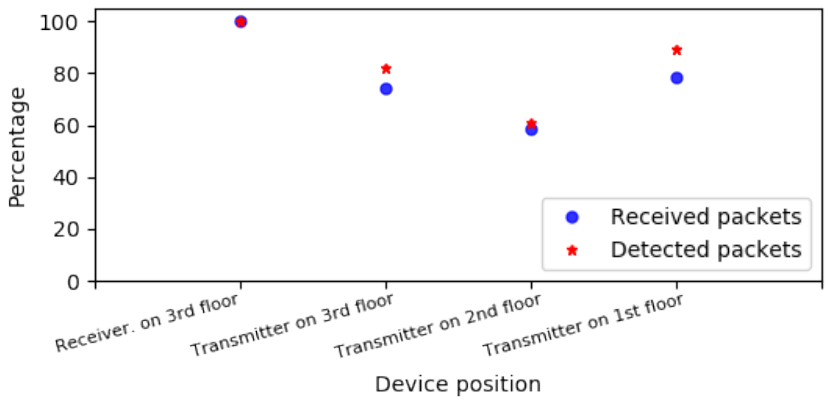

Figure 6.55: The reception rate as measured on the 1st stationary receiver positioned outside and on the 2nd stationary receiver the ("Receiver on the 3rd floor") during the measurement F2.

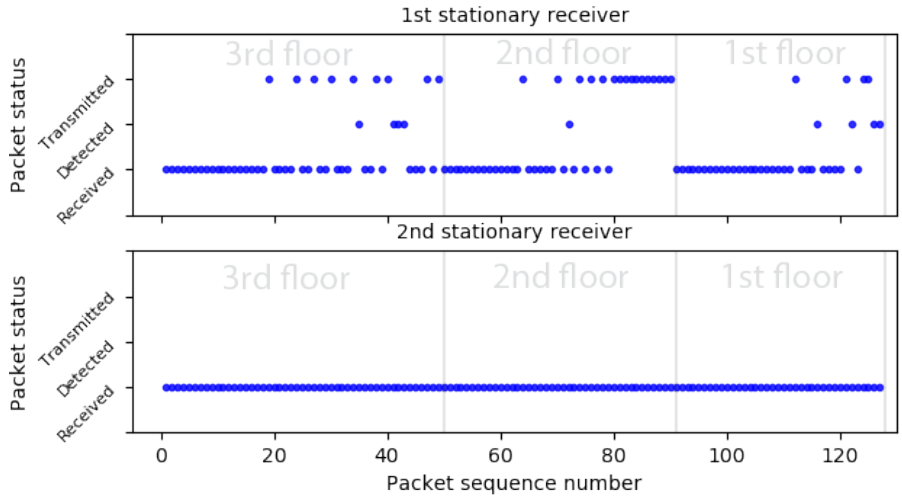

Figure 6.56: The transmitted, detected and received packets for all of the floors as measured on the 1st stationary receiver (top) and the 2nd stationary receiver (bottom) during the measurement F2. Each section with a floor number represents the position of the moving transmitter at that moment.

In the second part of the measurement F (the measurement F2), the reception rate is, in the respect to the measurement F1, almost the same on

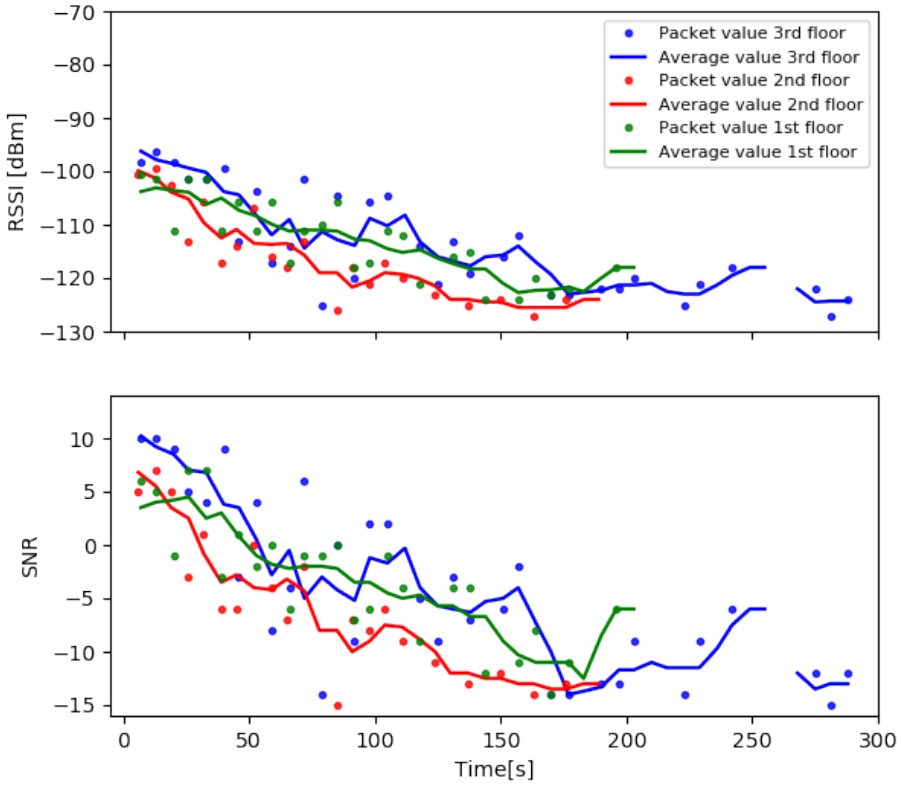

Figure 6.57: The RSSI and SNR values of the measurement F2 for all of the floors. The measurement on the third floor lasted for 260 seconds while the measurements on the other two floors lasted a little less (230 seconds).

all of the floors (the figures 6.55 and 6.56), while the RSSI and SNR values were a little higher on all of the floors especially in the second and third part of the building.

The visualization of the signal coverage through the building can be seen in figures 6.58 - 6.63.

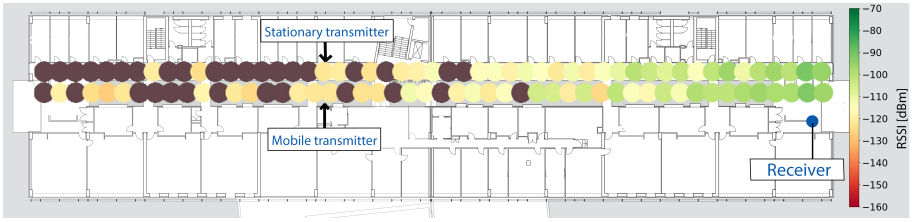

Figure 6.58: The RSSI values of both measurements (F1 and F2) for the third floor. The device on this figure denoted by the "Receiver" was in the case of the measurement F1 the control receiver and in the case of the measurement F2 the 2nd receiver.

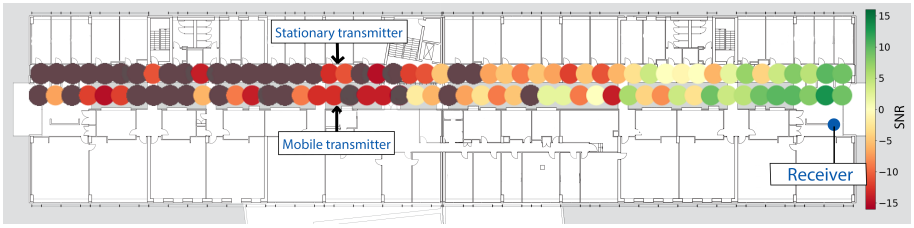

Figure 6.59: The SNR values of both measurements (F1 and F2) for the third floor. The device on this figure denoted by the "Receiver" was in the case of the measurement F1 the control receiver and in the case of the measurement F2 the 2nd receiver.

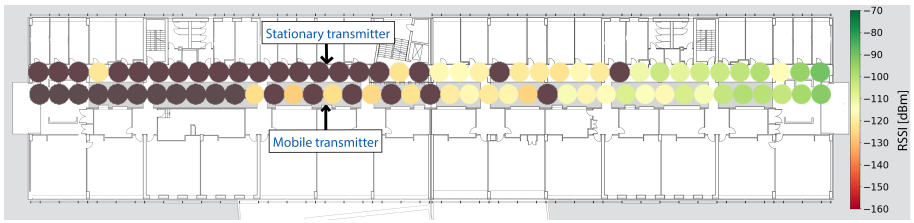

Figure 6.60: The RSSI values of both measurements (F1 and F2) for the second floor.

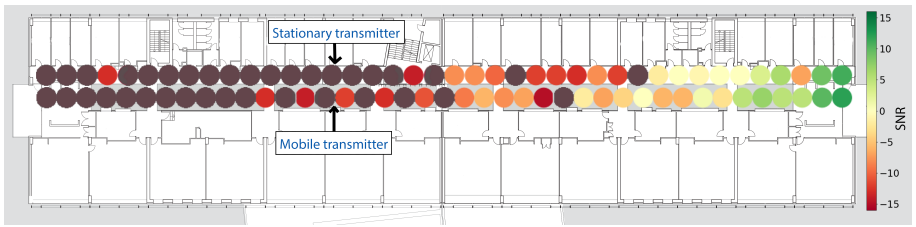

Figure 6.61: The SNR values of both measurements (F1 and F2) for the second floor.

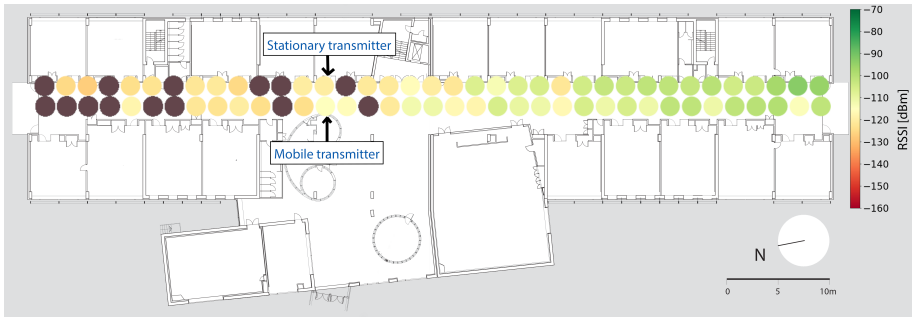

Figure 6.62: The RSSI values of both measurements (F1 and F2) for the first floor.

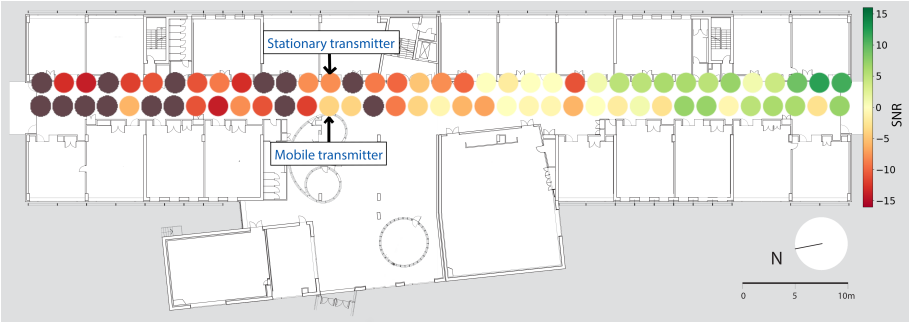

Figure 6.63: The SNR values of both measurements (F1 and F2) for the first floor.

6.2.4 Urban measurements

The next logical step in our testing was to conduct the tests where we measured the LoRa coverage in the urban area. For these measurements we decided to put one transceiver and the control receiver on top of the Rožnik hill in Ljubljana and drive around with another transceiver, placed on the the cars roof, in the city streets.

The place where the stationary transceiver and control receiver were positioned is 90 meters above the city streets where the measurements were conducted. Both devices were placed on a bench one meter above the ground and two meters apart. Their position on the hill (the red dot) and the stationary transceivers line of sight (the light blue shaded area on the map) can be seen in figure 6.64. The measurements for the same set of the LoRa parameter settings were repeated three times for a few days in a row on the same path trough the city (the blue colored path in figure 6.64). Because the part of the road was closed due to the construction work (the red path on map) and we wanted to measure the signal on a nice "circular" path, for easier analysis and interpretation, we had to conduct each measurement in two parts. In the first part, we drove from the fixed transceiver towards the point marked with number one and then, partially on the same path, to the point number two. At this time, when we were driving from the point number one

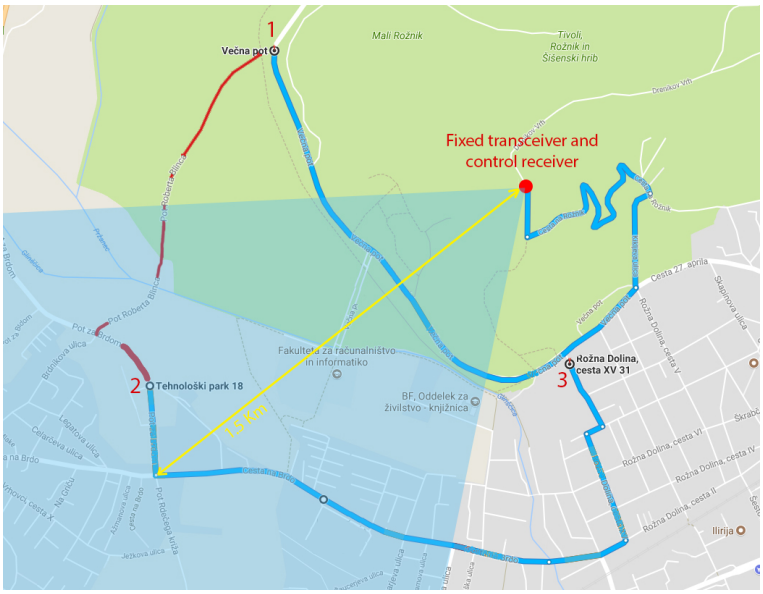

Figure 6.64: The path where urban measurements were conducted (colored in blue) and the line of sight area shaded with light blue.

towards the point number two, we did not record measurements. When point two was reached, we turned around, started recording the measurements and drove back to point three where we concluded with measurement. It is also worth noting that these measurements were done on the city roads with a light traffic and only a few traffic lights so the measurements duration differs a little.

For these measurements, we also used the bi-directional mode and the same settings as we did for the indoor and outdoor-indoor measurements at the Faculty of Computer and Information Science, described in two previous sections. We will denote the part of the measurements where the 250 kHz bandwidth was used as the "measurement G" and the part with the 125 kHz bandwidth as the "measurement H".

First measurement

We will, in the first part of the measurement G, discuss the measurements results where we had the stationary transmitter and the moving receiver. This part of the measurements will be denoted with the "measurement G1".

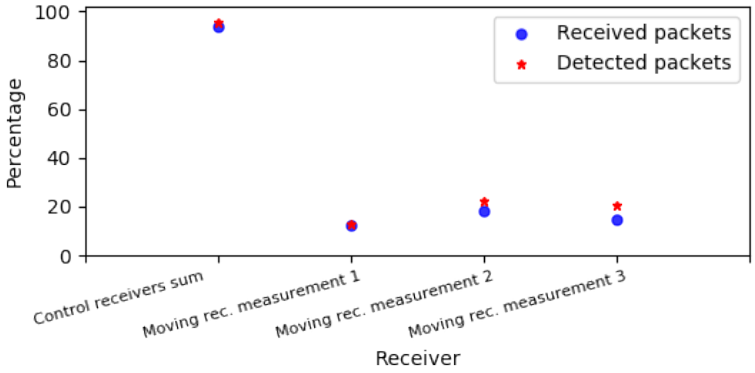

Figure 6.65: The detected vs. the received packet percentage measured on the control and moving receiver for all three measurements of the measurement G1. For the control receiver, the average of all three measurements is shown.

As one can see from figure 6.65, the detection and reception rates for the moving receiver are for all three measurements below 25% (the average detection rate of all three measurements is 18.8%, while the the average reception rate is 15%). The only exception is the control receiver, which was placed two meters away from the transmitter and has 94% reception rate (this reception rate is the sum of all three measurements). The main reason why the reception rates were so low lies in the LoRa parameter settings which are intended for a higher throughput rather than for transmitting data over the long distances. The other reason is probably the interference from the surrounding area. This can be seen from the control receiver measurements (figure 6.67), where the reception rate is a little lower in the first part of all three measurements (the packet sequence numbers below 30).

As we can see from figures 6.66 and 6.68, the packets were detected and received during all three measurements pretty much at the same time. If we

take into account that we drove with the same speed during each measurement, this means that the receptions occurred at the same places in the city. This conclusion also confirms the map in figure 6.69 which shows the places where the packets were received.

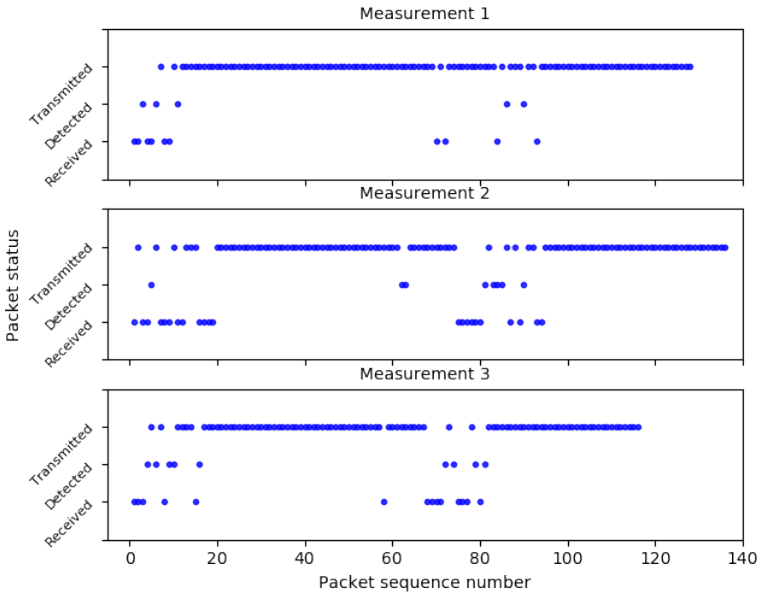

Figure 6.66: The transmitted, detected and received packets for all three repetitions of the measurement G1 as measured on the moving receiver.

A little more interesting are, in this case, the RSSI and SNR values. Figure 6.68 shows us that the signal was very strong in the areas with "line of sight" (there were only a few buildings in between the transmitter and the receiver), but it was completely lost when there were more obstacles (or a hill) between transmitter and receiver. The RSSI ans SNR measurements from the "line of sight" area can be seen on the plot in figure 6.68 between the 380th and 580th second.

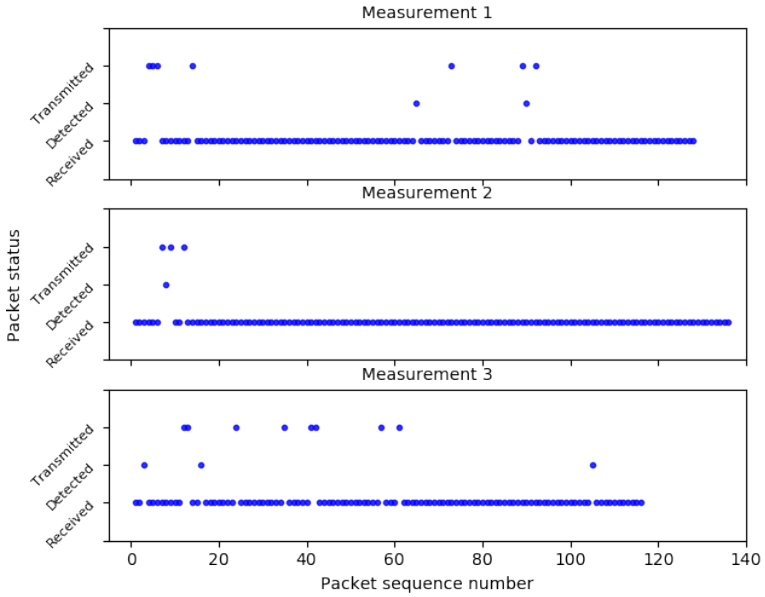

Figure 6.67: The transmitted, detected and received packets for all three repetitions of the measurement G1 as measured on the control receiver.

In the second part of the measurement G (measurement G2), where we had two stationary receivers (denoted as they were forthe measurements C2 and D2) and the moving transmitter, the results are very similar to the ones in the measurement G1 as are also the reception areas (figure 6.70). The average detection and reception rates were 27.3% and 20.8%, respectively. The reception rate was in each iteration of the measurement less than 10% higher (figures 6.72, 6.73 and 6.74), while the RSSI and SNR values were almost the same (figure 6.71).

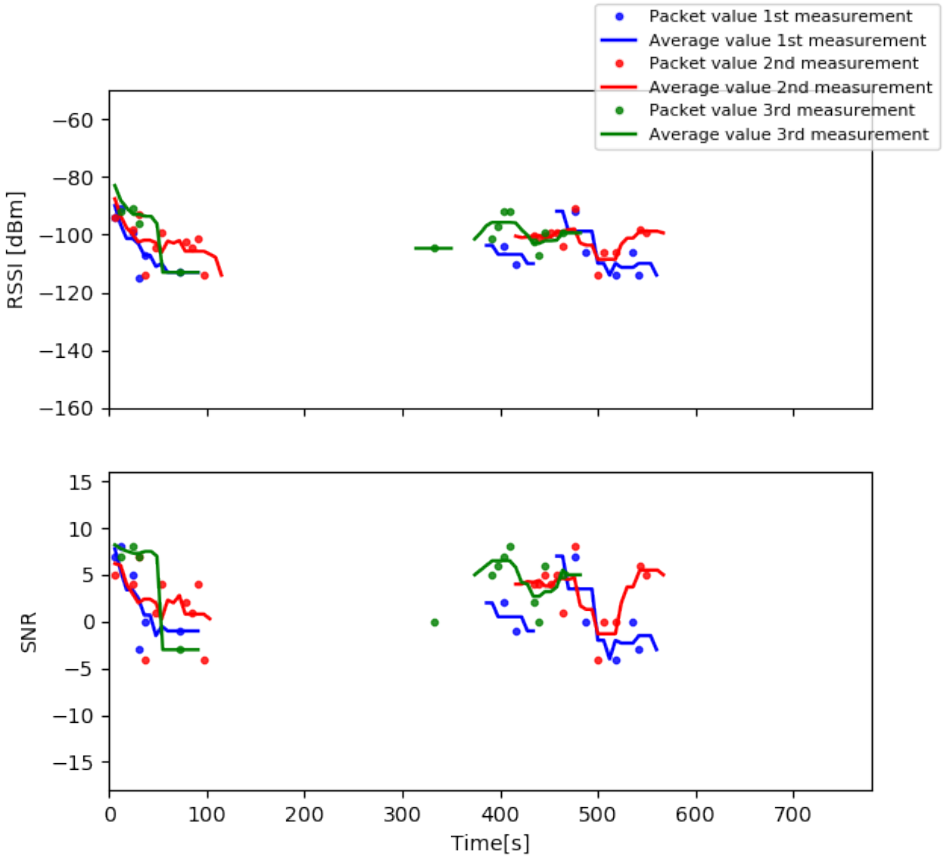

Figure 6.68: The RSSI and SNR values measured on the moving receiver during the measurement G1. The duration of all three measurements was around 700 seconds but there was no packet reception during the last 120 seconds of the measurements.

Figure 6.69: This map shows the area where the packets were received (the green, blue and yellow dots) and the places where there was no reception during the measurement G1.

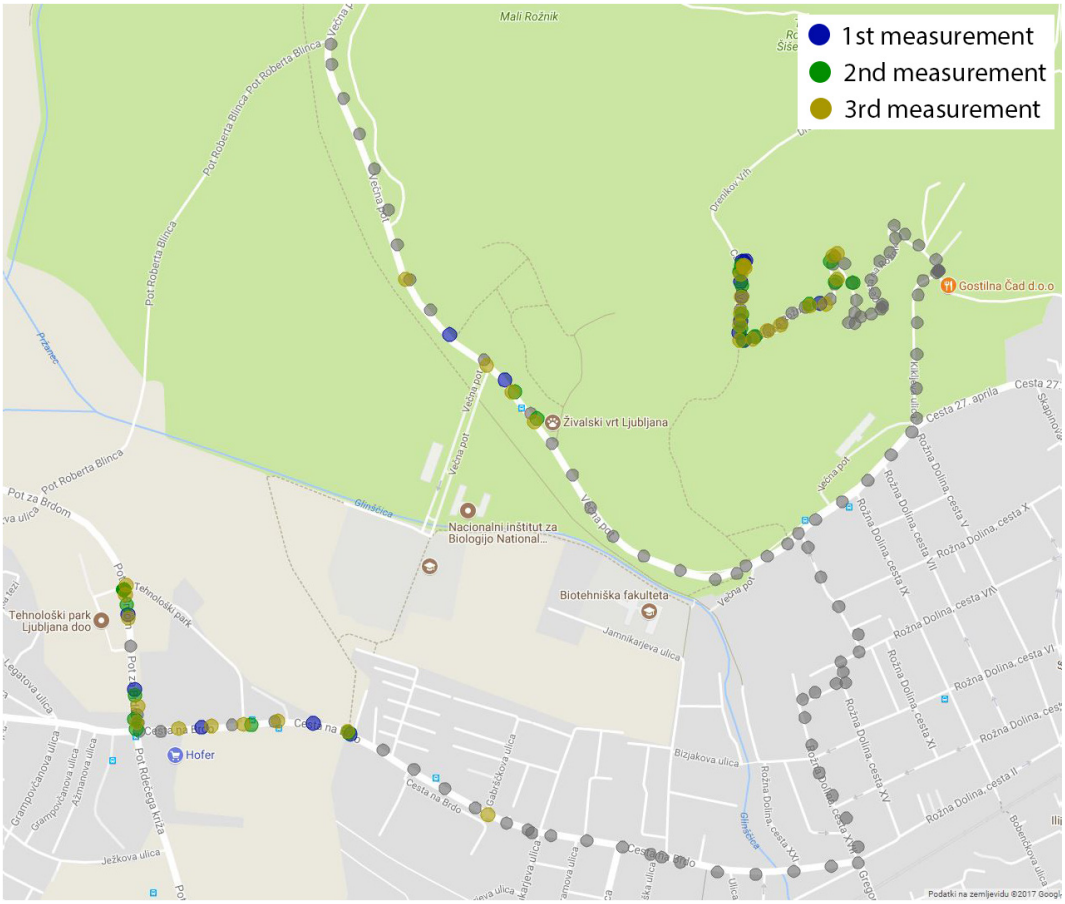

Figure 6.70: This map shows the area where the packets were received (the green, blue and yellow dots) and the places where there was no reception during the measurement G2.

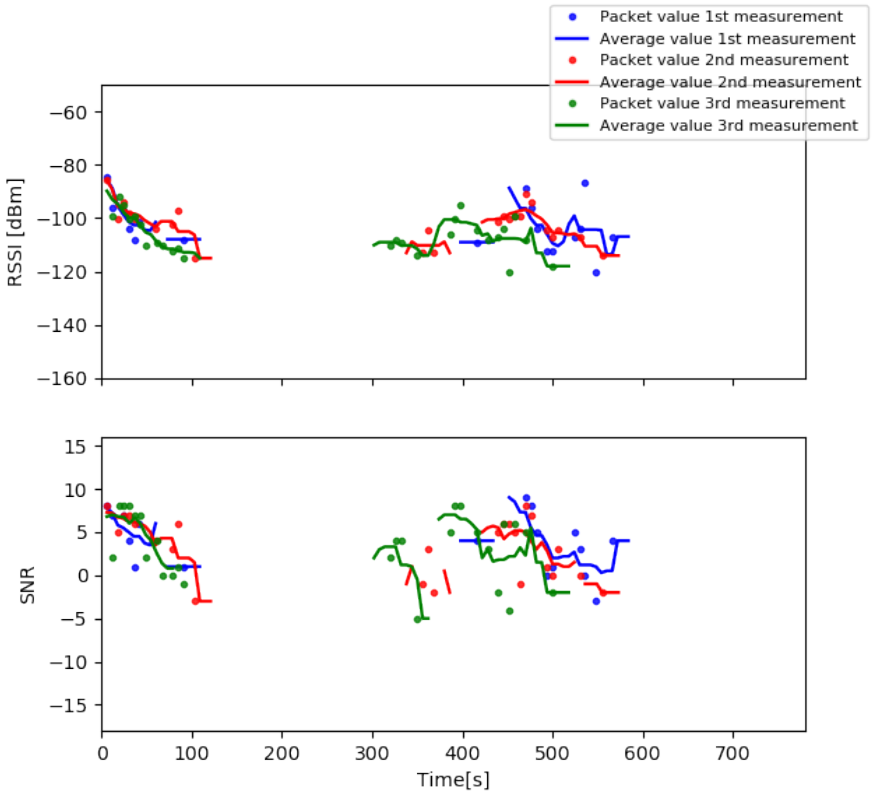

Figure 6.71: The RSSI and SNR values measured on the 1st stationary receiver during the measurement G2. The duration of all three measurements was around 700 seconds but there was no packet reception during the last 120 seconds of the measurements.

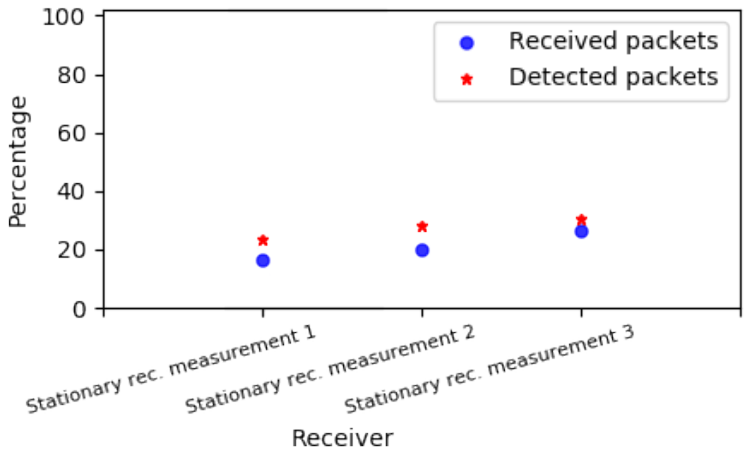

Figure 6.72: The detected vs. the received packet percentage measured on the 1st stationary receiver for all three measurements of the measurement G2.

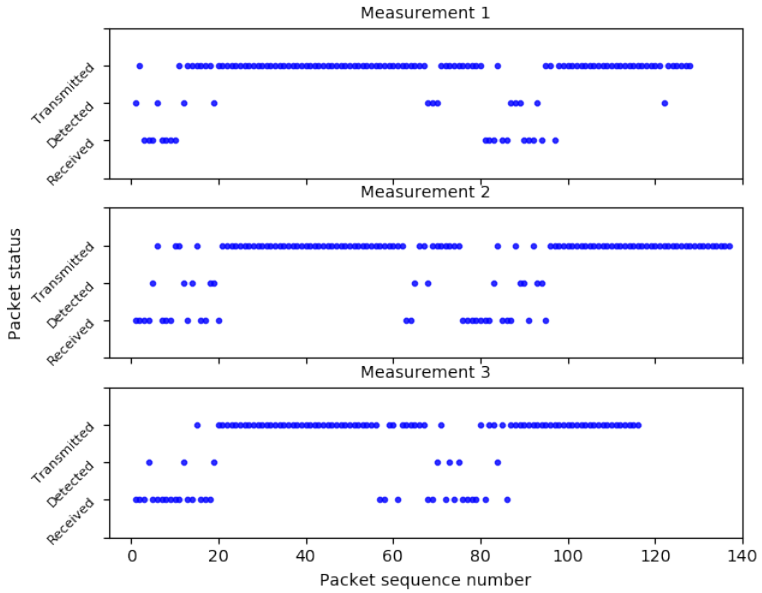

Figure 6.73: The transmitted, detected and received packets for all three repetitions of the measurement G2 as measured on the 1st stationary receiver.

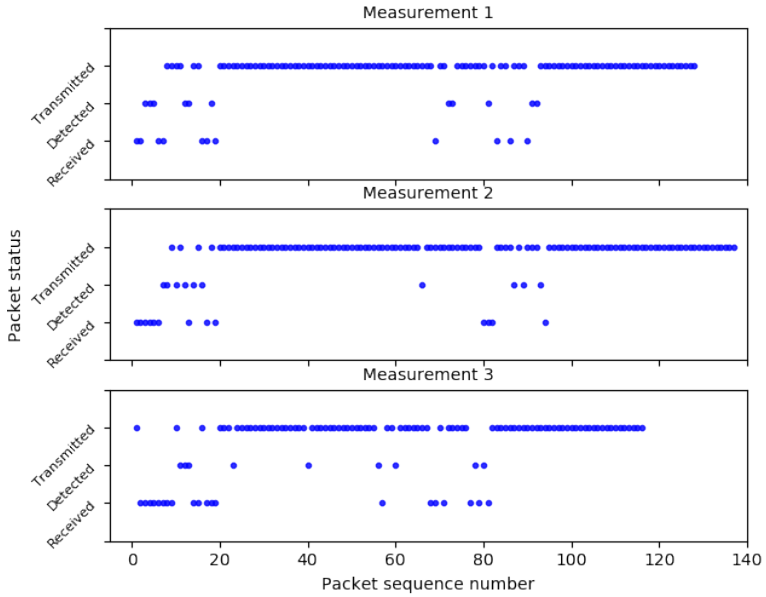

Figure 6.74: The transmitted, detected and received packets for all three repetitions of the measurement G2 as measured on the 2nd stationary receiver.

Second measurement

In this section, we will focus on the interpretation of the results from the measurement H where the 125 kHz bandwidth, the spreading factor 10 and the CRC turned on with the coding rate 4/6 were used. These settings are, in contrast to the ones fromthe measurement G, intended for transmitting data over longer distances while using a lower throughput.

As usual we will first look at the part of the measurement F where we had the stationary transmitter and the moving receiver. This part will be denoted by the measurement H1.

We can already see from figure 6.75 that better results were achieved with this set of parameter settings in comparison to the ones from the measurement G. The average detection and reception rates are in this measurement 47.7% and 35.9%, which is 30% and 20% higher than the ones in the measurement G1, respectively. To the higher reception rates probably also contributed the fact that during these measurements there was almost no

interference from the surrounding area (figure 6.78).

Similar to the measurements G, the packet reception occurred at the same area on the streets, as figure 6.76 shows, but this time, the area was much wider. By comparing the figure 6.66 from measurement G1 and figure 6.77 from this measurement, we can also see, that in this measurement the signal detection occurred even in a few other places in the city where in the measurement G1 it did not.

This time, the RSSI values (figure 6.79) are a little lower than the ones the from measurement E1, while the SNR values have a much bigger span of values (from +13 to -16 in comparison to the measurement G1 where the SNR spans from +9 to -5).

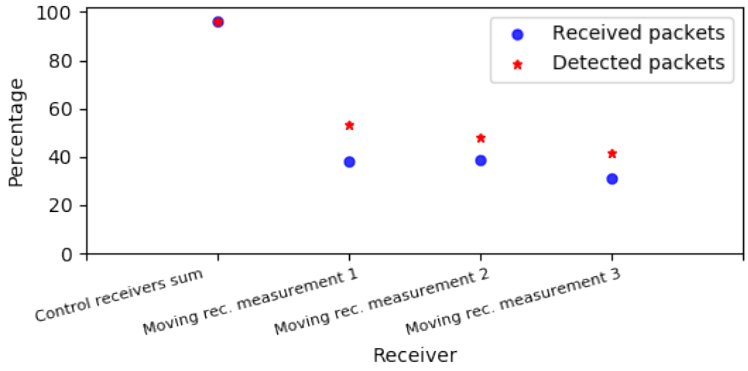

Figure 6.75: The detected vs. received packet percentage measured on the control and the moving receiver for all three repetitions of the measurement H1. For the control receiver, the average of all three measurements is shown.

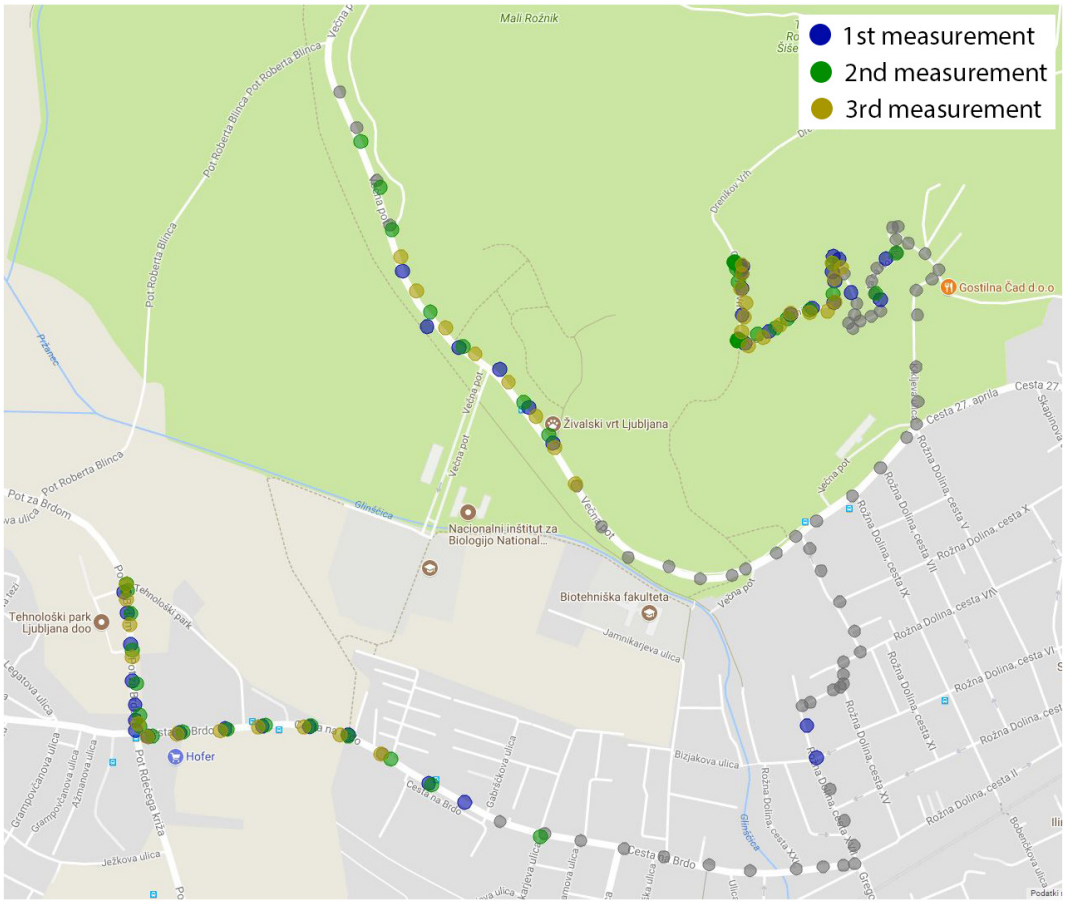

Figure 6.76: This map shows the area where the packets were received (the green, blue and yellow dots) and the places where there was no reception during the measurement H1.

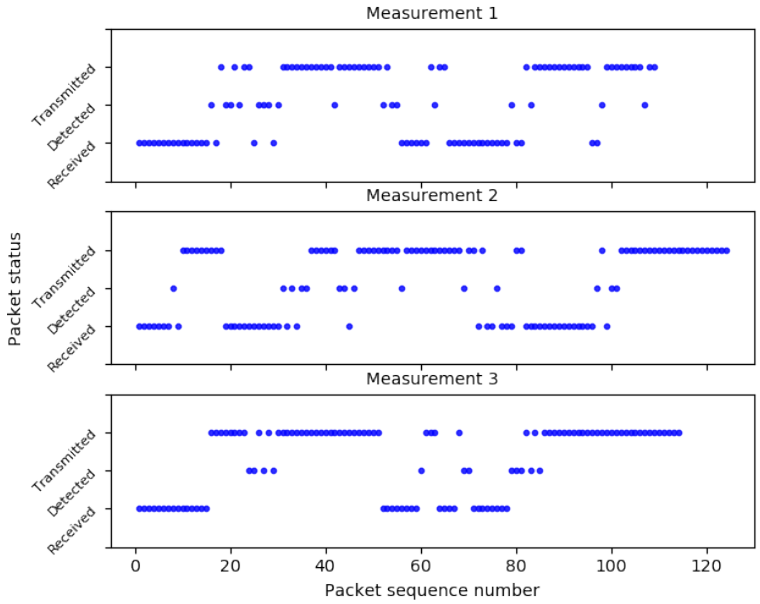

Figure 6.77: The transmitted, detected and received packets for all three repetitions of the measurement H1 as measured on the moving receiver.

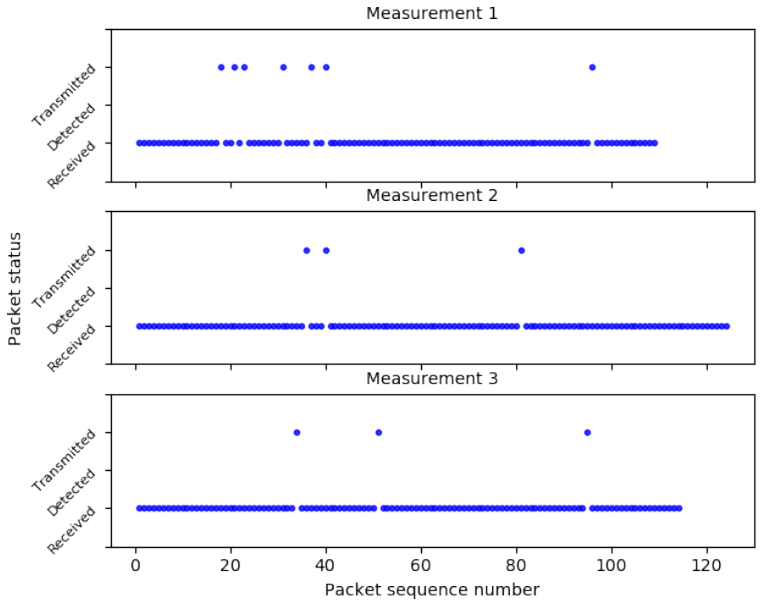

Figure 6.78: The transmitted, detected and received packets for all three repetitions of the measurement H1 as measured on the control receiver.

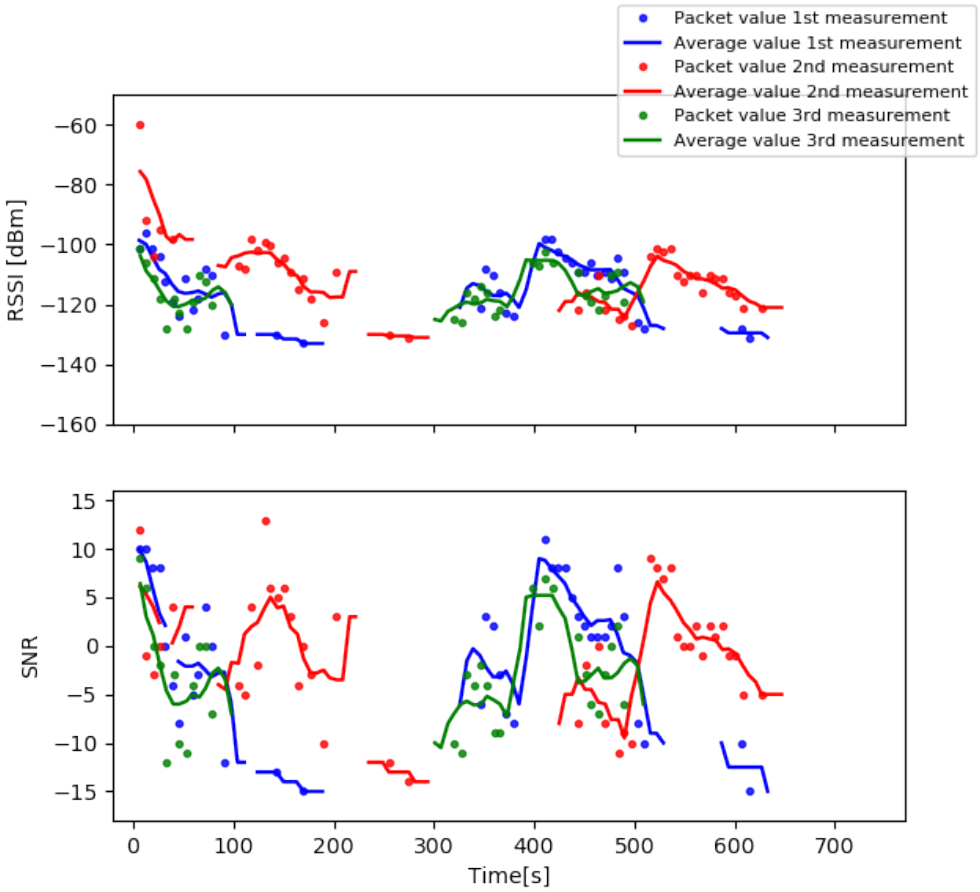

Figure 6.79: The RSSI and SNR values measured on the moving receiver during the measurement H1. The duration of all three measurements was around 700 seconds but there was no packet reception during the last 100 seconds of the measurements.

In the second part of the measurement H (the measurement H2), where we have the moving transmitter and two stationary receivers, denoted as for the measurement C2, the differences between these results and the ones from measurement H1 are very similar to the ones between the measurements G1 and G2. The average detection and reception rates are in this case 58.7% and 48.6%, respectively, which gives us a very similar difference in percentage (figures 6.80, 6.81 and 6.82). By comparing the plots in figures 6.79 and 6.83, we can see that RSSI and SNR values of the measurement H2 follow almost the same pattern as the ones in the measurement G2. The area where the packets were received can be seen in figure 6.84

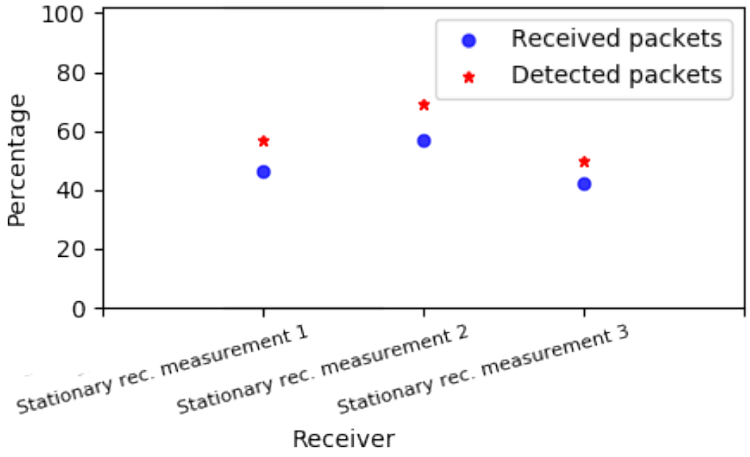

Figure 6.80: The detected vs. the received packet percentage measured on the 1st stationary receiver for all three measurements of the measurement H2.

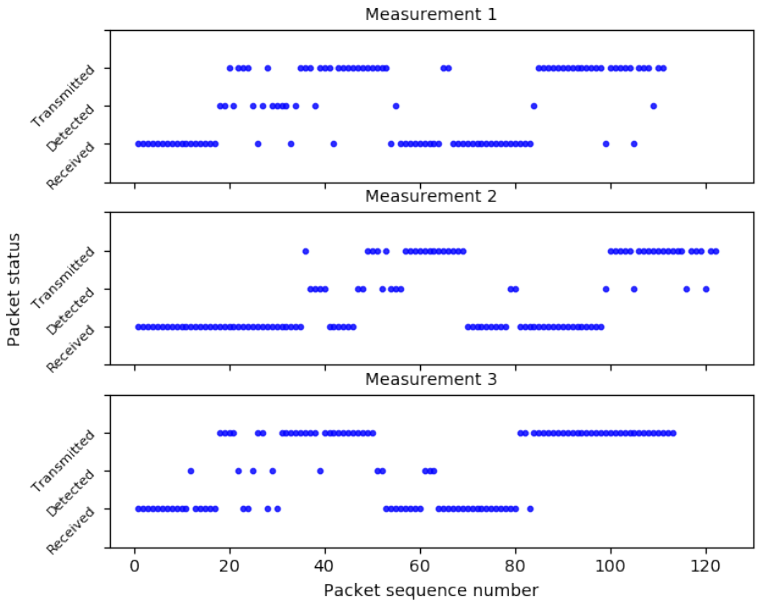

Figure 6.81: The ransmitted, detected and received packets for all three repetitions of the measurement H2 as measured on the 1st stationary receiver.

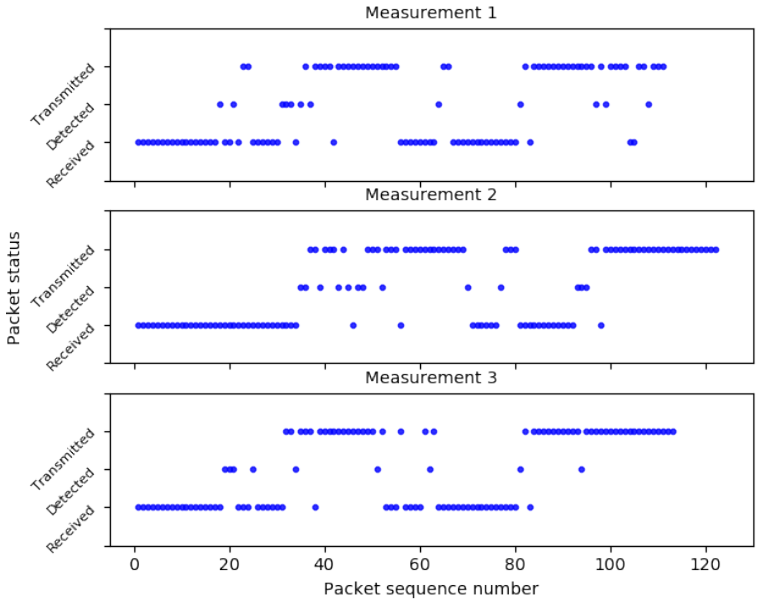

Figure 6.82: The transmitted, detected and received packets for all three repetitions of the measurement H2 as measured on the 2nd stationary receiver.

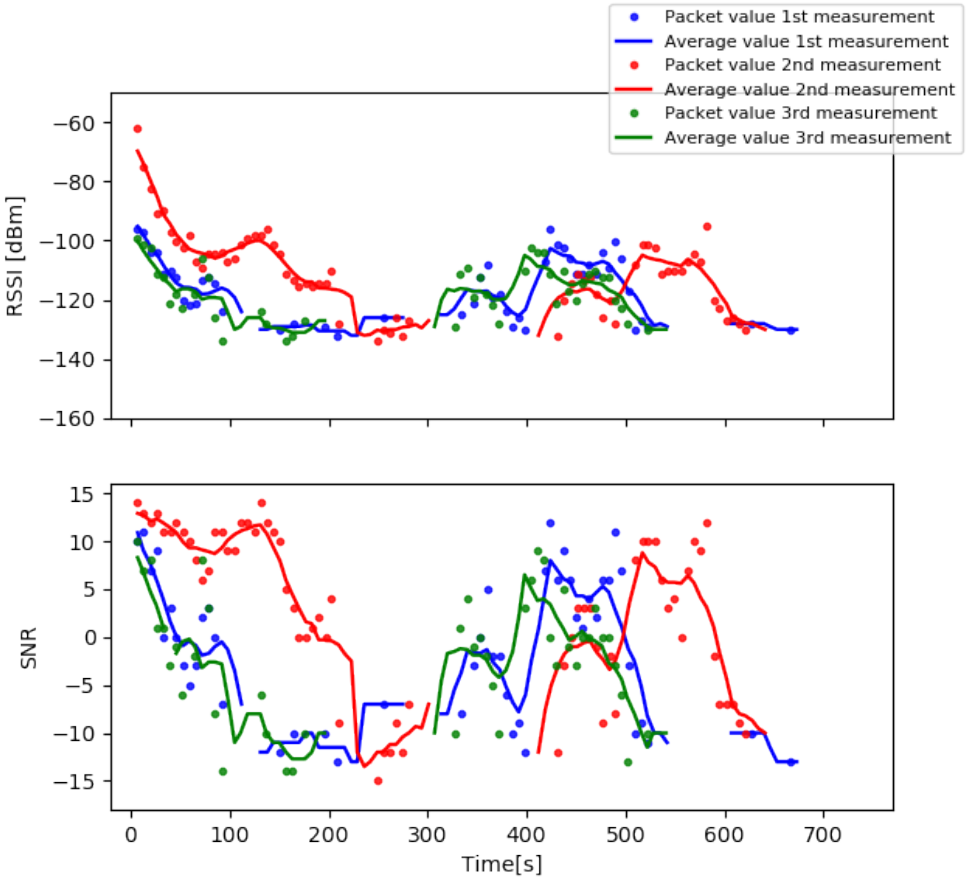

Figure 6.83: The RSSI and SNR values measured on the moving receiver during the measurement H2. The duration of all three measurements was around 700 seconds but there was no packet reception during the last 100 seconds of the measurements.

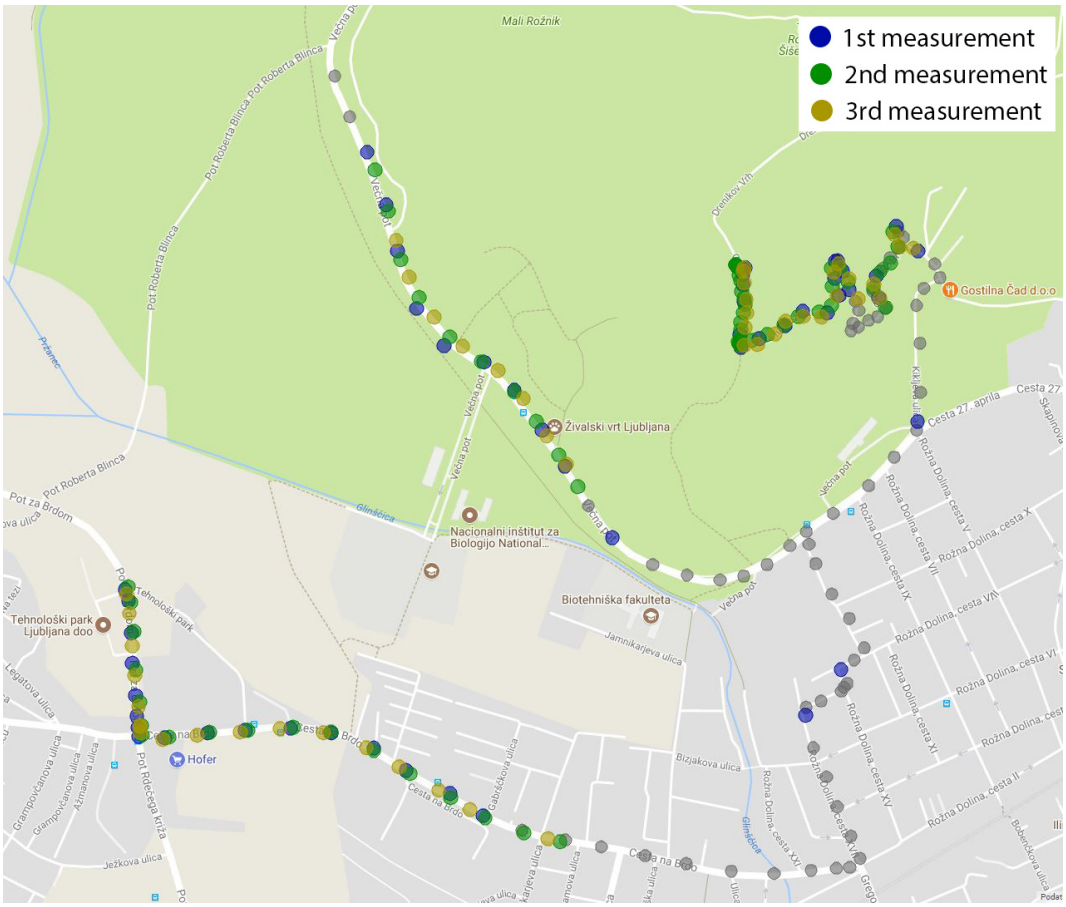

Figure 6.84: This map shows the area where the packets were received (the green, blue and yellow dots) and the places where there was no reception during the measurement H2.

Chapter 7

Conclusion

With this master's thesis we have successfully delivered a flexible and scalable open source research platform which is free to use and available on GitHub [42]. This platform currently supports only the Rapberry Pis 3 Model B, the Semtechs SX1276 LoRa chips and the Android smart phones, but it can be upgraded to support all kinds of devices and chips. It is easy to use, but most importantly, all devices connected to this platform can be controlled from a single place. The commands and settings for all of the devices can be stored in a single JSON file which is multicasted using the single Unix terminal command. If one wants to have control over all of the devices from a single place, they all have to be connected to the IP network, otherwise, each individual device has to be configured manually.

We also conducted a few LoRa signal range, coverage and penetration measurements. The best measurement result worth mentioning is the one from the range test conducted during the preliminary testings of our research platform. For this test, the transmitter was placed on the mountain (1320 meters above the sea level), while we drove, with the receiver placed on top of the cars roof, downhill and away from the mountain. Both devices used for this test were identical - the Raspberry Pi 3 Model B with the custom LoRa module and 1/4 wave whip antenna. The range achieved during this test was 39 kilometers with a line of sight between the devices. This is, to our

knowledge, the longest distance achieved on land, using the LoRa modulation and no other paper (as far as we know)has reported similar results up to date.

Our measurements presented in this work also show how a different packet reception rate can be achieved by choosing a different placement of the devices or by changing the LoRa parameter settings. They also show us that if we have the bi-directional communication between the devices, the signal strength, the packet detection and the packet reception rates differ in the direction from the stationary transmitter to the moving receiver from the ones in the other direction. During all of our measurements the packet detection and reception rates were nearly 10% higher in the part of the measurements where we had the moving transmitter and the stationary receiver (the uplink communication), than vice versa. The reason for this constant difference could be the fact that we always used (for all of the tests) the same devices as the stationary and as the moving transceiver. Unfortunately, we did not have enough time to test, if this is in fact the consequence of using the same devices for the stationary or the moving transceiver.

Overall, the results of our measurements indicate that the LoRa

- coverage is basically associated to the presence of the LOS (or good diffracted paths) in the long outdoor-outdoor links,
- reception is still good in the indoor-indoor links,
- for the outdoor-to-indoor and the indoor-to-outdoor communication seems to be rather problematic.

During the process of making this master's thesis (during the development and during the conducting measurements), we learned that it is very important to plan in details how you are going to do something. Despite the good planing, sometimes, something can go wrong. We also learned that it is not easy to achieve a good urban coverage, despite the fact that LoRa has a great range. If one wants to achieve a good coverage, they have to either set the LoRa modulation parameters so that they have a very low bandwidth or place the devices (and the base stations) as high as possible.

The most important thing to do, in the further work on this platform, is to correct the devices synchronization bug when using the bi-directional mode. Now, with this bug present, the bi-directional mode can only be used if the time between two transmissions is greater than 3000 milliseconds, otherwise the synchronization between the devices is lost. The further work also includes the testing if the difference in the packet detection and reception rates is the consequences of using the same stationary and moving device for all of the measurements or it is due to some other properties. An upgrade on this platform could also be made to support the multiple kinds of the end-devices and different LoRa chips, while the GPS receiver could be integrated on end-devices, which will eliminate the need of the Android application while conducting the outdoor measurements.

Bibliography

[1] M. Centenaro and L. Vangelista and A. Zanella and M. Zorzi, "Long-Range Communications in Unlicensed Bands: the Rising Stars in the IoT and Smart City Scenarios," *in IEEE Wireless Communications*, vol. 23, no. 5, October 2016, pp. 60-67.

[2] LoRa Alliancem, `https://www.lora-alliance.org/`, accessed: 2017-07-09.

[3] Github - rpp0/gr-lora, `https://github.com/rpp0/gr-lora`, accessed: 2017-08-04.

[4] Github - BastilleResearch/gr-lora, `https://github.com/BastilleResearch/gr-lora`, accessed: 2017-08-04.

[5] Github - myriadrf/LoRa-SDR, `https://github.com/myriadrf/LoRa-SDR`, accessed: 2017-08-04.

[6] J. Petäjäjärvi, K. Mikhaylov, M. Hämäläinen and J. Iinatti, "Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring," *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, Worcester, MA, 2016, pp. 1-5.

[7] P. Neumann, J. Montavont and T. Noël, "Indoor deployment of low-power wide area networks (LPWAN): A LoRaWAN case study," *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, New York, NY, 2016, pp. 1-8.

[8] 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, *IEEE Wireless Communications*, Vol. 23, Oct. 2016.

[9] L. Gregora, L. Vojtech and M. Neruda, "Indoor signal propagation of LoRa technology," 2016 17th International Conference on Mechatronics - Mechatronika (ME), Prague, 2016, pp. 1-4.

[10] T. Petrić, M. Goessens, L. Nuaymi, L. Toutain and A. Pelov, "Measurements, performance and analysis of LoRa FABIAN, a real-world implementation of LPWAN," *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Valencia, 2016, pp. 1-7.

[11] A. Augustin, J. Yi, T. Clausen, W. M. Townsley, "A Study of LoRa: Long Range &; Low Power Networks for the Internet of Things," *Sensors 2016*, 16(9), page 1466.

[12] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia and N. Strachan, "Evaluation of LoRa and LoRaWAN for wireless sensor networks," *2016 IEEE SENSORS*, Orlando, FL, 2016, pp. 1-3.

[13] D. H. Kim, J. B. Park, J. H. Shin and J. D. Kim, "Design and implementation of object tracking system based on LoRa," *2017 International Conference on Information Networking (ICOIN)*, Da Nang, 2017, pp. 463-467.

[14] W. San-Um, P. Lekbunyasin, M. Kodyoo, W. Wongsuwan, J. Makfak and J. Kerdsri, "A long-range low-power wireless sensor network based on U-LoRa technology for tactical troops tracking systems," *2017 Third Asian Conference on Defence Technology (ACDT)*, Phuket, 2017, pp. 32-35.

[15] L. Li, J. Ren and Q. Zhu, "On the application of LoRa LPWAN technology in Sailing Monitoring System," *2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Jackson, WY, 2017, pp. 77-80.

[16] K. Mikhaylov, .. Juha Petaejaejaervi and T. Haenninen, "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology," *European Wireless 2016; 22th European Wireless Conference*, Oulu, Finland, 2016, pp. 1-6.

[17] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, J. Iinatti, "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage," *International Journal of Distributed Sensor Networks*, 2017, Vol. 13(3), 2017, pp. 1-16.

[18] F. Adelantado, X. Vilajosana, P. Tuset, B. Martinez, J. Melia-Segui, et al.. "Understanding the Limits of LoRaWAN," *IEEE Communications Magazine*, Institute of Electrical and Electronics Engineers, 2017.

[19] D. Bankov, E. Khorov and A. Lyakhov, "On the Limits of LoRaWAN Channel Access," *2016 International Conference on Engineering and Telecommunication (EnT)*, Moscow, 2016, pp. 10-14.

[20] J. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hanninen and M. Pettissalo, "On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology," *2015 14th International Conference on ITS Telecommunications (ITST)*, Copenhagen, 2015, pp. 55-59.

[21] O. Georgiou and U. Raza, "Low Power Wide Area Network Analysis: Can LoRa Scale?," *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162-165, April 2017.

[22] M. Haghighi, Z. Qin, D. Carboni, U. Adeel, F. Shi and J. A. McCann, "Game theoretic and auction-based algorithms towards opportunistic communications in LPWA LoRa networks," *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, VA, 2016, pp. 735-740.

[23] F. Orfei, C. Benedetta Mezzetti and F. Cottone, "Vibrations powered LoRa sensor: An electromechanical energy harvester working on a real bridge," *2016 IEEE SENSORS*, Orlando, FL, 2016, pp. 1-3.

[24] J. So, D. Kim, H. Kim, H. Lee and S. Park, "LoRaCloud: LoRa platform on OpenStack," *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, Seoul, 2016, pp. 431-434.

[25] M. Loy and R. Karingattil and L. Williams, "ISM-Band and Short Range Device Regulatory Compliance Overview," *Texas instruments*, 2015.

[26] Semtech Corporation, "SX1272/3/6/7/8: LoRa modem design guide. AN1200.13," Revision 1, 2013.

[27] "Measurement of distances without a roulette and wires", `http://developers-club.com/posts/153237/`, accessed: 2017-06-11.

[28] CubicSRD, Cross-platform and open-source software-defined radio application, `https://cubicsdr.com/`, accessed: 2017-06-11.

[29] RTL-SDR.COM, `http://www.rtl-sdr.com/`, accessed: 2017-06-11.

[30] Semtech Corporation, "LoRa modulation basics. AN1200.22," Revision 2, 2015.

[31] Semtech Corporation, "SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver," Revision 4, 2015.

[32] Alexandros-Apostolos A. Boulogeorgos, Panagiotis D. Diamantoulakis, George K. Karagiannidis, "Low Power Wide Area Networks (LPWANs) for Internet of Things (IoT) Applications: Research Challenges and Future Trends,", 2016, pp. 1-16.

[33] N. Sornin (Semtech) and M. Luis (Semtech) and T. Eirich (IBM) and T. Kramp (IBM) and O.Hersent (Actility), "LoRaWAN™ Specification,", V: 1.0.2, 2016.

[34] Semtech Corporation, "AN1200.23 Recommended SX1272 Settings for EU868 LoRaWAN Network Operation," Revision 1, 2015.

[35] Sigfox, https://www.sigfox.com/, accessed: 2017-06-11.

[36] Comparison of LPWAN Technologies – Which is Best For Me?, https://iot-for-all.com/comparison-of-lpwan-technologies/, accessed: 2017-06-11.

[37] Link Labs: Low power wide area networks (LPWAN) and LTE-M fot IoT, https://www.link-labs.com/, accessed: 2017-06-11.

[38] Ingenu - Dedicated machine connectivity for IoT, https://www.ingenu.com, accessed: 2017-06-11.

[39] Weightless - Setting the standard of IoT, https://www.weightless.org, accessed: 2017-06-11.

[40] MQTT, http://mqtt.org/, accessed: 2017-06-11.

[41] MQTT, http://www.eclipse.org/paho/clients/python/docs/, accessed: 2017-07-14.

[42] misterGrosar/LoRaTestPlatform, https://github.com/misterGrosar/LoRaTestPlatform, accessed: 2017-08-23.