DTU Library

Evaluation of Cellular IoT Technologies for Critical Applications

Ballal, Kalpit Dilip

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

DTU Electro
Department of Electrical and Photonics Engineering

Evaluation of Cellular IoT Technologies for Critical Applications

PhD Thesis

Evaluation of Cellular IoT Technologies for Critical Applications

PhD Thesis
July, 2023

By
Kalpit Dilip Ballal

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Electrical and Photonics Engineering, Building 343, Ørsteds Plads, 2800 Kgs. Lyngby Denmark
https://electro.dtu.dk
ISSN: [0000-0000] (electronic version)
ISBN: [000-00-0000-000-0] (electronic version)

ISSN: [0000-0000] (printed version)
ISBN: [000-00-0000-000-0] (printed version)

Approval

This thesis has been prepared at the Department of Electrical and Photonics Engineering together as a part Network Technologies and Services Platforms group at Technical University of Denmark.

Kalpit Dilip Ballal

Kalpit Dilip Ballal

....

Signature

31/07/2023

....

Date

Abstract

The world is experiencing a new wave of connected devices that can measure temperature, humidity, water levels, etc., and the volume of these devices is increasing exponentially over time. The concept of Internet of Things (IoT) existed for several years, but in recent years, the introduction of Low Power Wide Area Network (LPWAN) technologies has expanded its scope of applications in several other verticals. Now the use of IoT devices can be found in industrial automation applications, utilities, smart cities, smart communities, transportation, etc. One vertical that can benefit from using the new generation of LPWAN technologies is remote healthcare. This thesis validates the use of Cellular Internet of Things (C-IoT) technologies for building an End-to-End (E2E) critical application that is expected to perform continuous remote monitoring of chronically ill Cardiovascular Disease (CVD) patients. The project follows Participatory Design methodology to design an IoT end device that is capable of performing continuous measurement of Electrocardiogram (ECG) and Heart Rate (HR). The device can then send that data to a remote data collection, storage, and visualization system using C-IoT technologies. The efforts in evaluating the KPI performance of the C-IoT technologies involved performing an experimental evaluation of network Key Performance Indicator (KPI) such as E2E latency, bitrate, and Packet Drop (PD) in several physical environments. The experiments were performed in indoor, deep-indoor, outdoor, remote-outdoor, and network roaming environments. The project further goes on to evaluate the C-IoT per cell capacity and the power consumption recorded using different C-IoT modules to understand the wholistic picture of the state of C-IoT network deployed by Denmark's two biggest MNO.

The outcome of the experimental evaluation of C-IoT networks in various conditions perform differently. There have been noticeable KPI performance dissimilarities in the different MNO within the country and in the Nordic region. The Radio Access Network (RAN) parameter settings can affect the capacity of the number of devices that can perform continuous data transmission as well the RAN settings can affect the power consumption of the C-IoT devices. The experiment also resulted in testing C-IoT devices in an outage scenario and discovered the poor performance capabilities of these devices to maintain connectivity in outage situations. This highlights the limitation of using a single frequency band to deploy the C-IoT network. Identification of gaps in the current C-IoT deployments led to using different techniques such as Multiple Radio Access Technology (Multi-RAT) and Device-TO-Device (D2D) communication to avoid complete loss of connectivity of IoT devices. During the project, a new prototype was developed that could combine these two approaches to enable more reliable and robust communication between the IoT device and a remote server.

The results from the experimental testing of C-IoT indicate that the technologies are not fully matured yet to support the deployment of critical IoT applications requiring continuous remote monitoring. Perhaps there is a need for a substandard that can allow these technologies to support continuous critical monitoring applications such as telemedicine and telemonitoring.

Resumé

Verden oplever en ny bølge af forbundne enheder, der kan måle temperatur, luftfugtighed, vandstand osv., og antallet af disse enheder stiger eksponentielt over tid. Konceptet Internet of Things (IoT) har eksisteret i flere år, men i de seneste år har introduktionen af Low Power Wide Area Network (LPWAN) teknologier udvidet dens anvendelsesområde inden for flere andre vertikaler. Nu kan brugen af IoT-enheder findes i industrielle automationsanvendelser, forsyningsvirksomheder, smarte byer, smarte fællesskaber, transport osv. En vertikal, der kan drage fordel af at bruge den nye generation af LPWAN-teknologier, er fjernsundhed. Denne afhandling validerer brugen af Cellular Internet of Things (C-IoT) teknologier til at opbygge en End-to-End (E2E) kritisk applikation, som forventes at udføre kontinuerlig fjernovervågning af kronisk syge kardiovaskulære sygdom (CVD) patienter. Projektet følger metoden Participatory Design til at designe en IoT-end-enhed, der er i stand til at udføre realtidsmåling af elektrokardiogram (ECG) og hjertefrekvens (HR) og sende disse data til et fjernt datasamlings-, opbevarings- og visualiseringssystem ved hjælp af C-IoT-teknologier. Anstrengelserne i at evaluere KPI-præstationen af C-IoT-teknologierne omfattede udførelse af en eksperimentel evaluering af netværksnøglepræstationsindikatorer (KPI) som f.eks. E2E-latens, bitrate og pakettab (PD) i flere fysiske miljøer. Eksperimenterne blev udført indendørs, dybt indendørs, udendørs, fjerntliggende udendørs og roaming-miljøer. Projektet fortsætter med at evaluere C-IoT pr. cellekapacitet og strømforbrug, der er registreret ved hjælp af forskellige C-IoT-moduler for at forstå det helhedsbillede, der gælder for C-IoT-netværket, der er implementeret af Danmarks to største mobilnetværksoperatører.

Resultatet af den eksperimentelle evaluering af C-IoT-netværk under forskellige forhold viser forskellige præstationer. Der har været mærkbare forskelle i KPI-præstationen på de forskellige mobilnetværksoperatører inden for landet samt i Norden som helhed. Indstillingerne for det radioadgangsnetværk (RAN) kan påvirke kapaciteten af antallet af enheder, der kan udføre kontinuerlig dataoverførsel, samt RAN-indstillingerne kan påvirke strømforbruget for C-IoT-enhederne. Eksperimentet resulterede også i test af Cellular Internet of Things (C-IoT) enheder i en nedbrudsscenarie og afslørede de dårlige præstationsmuligheder for disse enheder til at opretholde forbindelse i sådanne situationer, hvilket afslører begrænsningen ved at bruge en enkelt frekvens til at drive C-IoT-netværket. Identifikationen af sådanne huller i de nuværende C-IoT-implementeringer førte til brug af forskellige teknikker som f.eks. Multiple Radio Access Technology (Multi-RAT) og Device-TO-Device (D2D) kommunikation. I løbet af projektet blev der udviklet en ny prototype, der kunne kombinere disse to tilgange for at muliggøre mere pålidelig og robust kommunikation mellem IoT-enheden og en fjernserver.

Resultaterne fra den eksperimentelle test af C-IoT indikerer, at teknologierne endnu ikke er fuldt udviklede til at understøtte implementeringen af kritiske IoT-applikationer, der kræver kontinuerlig fjernovervågning. Måske er der behov for en standard, der kan tillade disse teknologier at understøtte kontinuerlig overvågning af kritiske applikationer som fjernsundhed.

w

Acknowledgements

First and foremost, I would like to thank my parents for their continuous support and encouragement throughout my journey in Denmark.

I would like to thank my supervisors, Professor Lars Dittmann, Professor Birthe Dinesen and Associate Professor Sarah Ruepp for their guidance, feedback, and for the opportunity to work on this interesting field technology. I would especially like to thank Lars for all help with things inside and outside of the boundaries of the university. I would like to thank Sarah not only for her supervision but also for the countless opportunities to collaborate on acquiring research grants, project meetings, and countless other things. I would like to thank Birthe Dinesen and Knud Larsen for their feedback, and support in understating the unknown field of clinical science and Telemedicine.

Many thanks to my peers Radheshyam Sing, Krzysztof Mateusz Malarski, Eder Ollora Zaballa, Mingyuan Zang, and Stanley Chukwuebuka Nwabuona for collaborations, lunch room discussions, and creating a home like atmosphere. I would also like extended my gratitude towards Kartikeya Sharma for helping me with the design of front page for the thesis. I would like to thank Radheshyam Sing, and Eder Ollora Zaballa for proofreading the thesis. A big thanks to all the Masters students at DTU Electro with whom I got the privilege to collaborate on a number of interesting topics.

I would like to express my gratitude towards all the Network Technologies and Service Platform group colleagues, Associate Professor Henrik Wessing, Michael Berger, Martin Nordal Petersen, and Lars Staalhagen.

I would also like to thank Jens-Peter Meesenburg for introducing me to the field of Internet of Things from a commercial angle and letting me be part of Denmark's first ever commercially deployed IoT network project.

Finally I would like to thank people who made my life enjoyable in Denmark, my friends and my partner Vasudha for the continued support and motivation.

List of Contents

List of Figures

List of Tables

Abbreviations

2G	Second-Generation Cellular Network.
3G	Third-Generation Cellular Network.
3GPP	3rd Generation Partnership Project.
5G NR	5G New Radio.
ACL	Asynchronous Connection-oriented Logical transport.
API	Application Programming Interface.
APN	Access Point Name.
BLE	Bluetooth Low Energy.
BPSK	Binary Phase Shift Keying.
C-IoT	Cellular Internet of Things.
cDRX	Connect mode Discontinuous Reception.
CNN	Convolutional Neural Networks.
CoAP	Constrained Application Protocol.
CSS	Chirp Spread Spectrum.
CVD	Cardiovascular Disease.
D2D	Device-TO-Device.
DL	DownLink.
DNN	Deep Neural Network.
DTU	Technical University of Denmark.
E-CID	Enhanced Cell ID.
E2E	End-to-End.
EC	Extended Coverage.
ECG	Electrocardiogram.
eDRX	Extended Discontinuous Reception.
EEG	Electroencephalogram.
eNB	Evolved Node B.
EPC	Evolved Packet Core.

EPS	Evolved Packet System.
FOTA	Firmware Over-The-Air.
FPT	Future Patient Telerehabilitation.
GDPR	General Data Protection Regulation.
GNSS	Global Navigation Satellite System.
GPIO	General Purpose Input/Output.
GSM	Global System for Mobile Communications.
GSM-R	Global System for Mobile Communications - Railway.
GSMA	Global System for Mobile communications Association.
GUI	Graphical User Interface.
HEX	Hexadecimal.
HF	Heart Failure.
HR	Heart Rate.
HTTP	Hypertext Transfer Protocol.
HTTPS	Hypertext Transfer Protocol Secure.
I2C	Inter-Integrated Circuit.
ICT	Information and Communications Technology.
IoT	Internet of Things.
JD TeleTech	Japanese & Danish Research Network on Telehealth/Telerehabilitation and Welfare Technologies.
Kbps	Kilobits per Second.
KNN	k-Nearest Neighbors.
KPI	Key Performance Indicator.
Li-ion	lithium-ion.
LOCATE	A LoRa-based Mobile Emergency Management System.
LoRa	Long Range.
LoRaWAN	Long Range Wide Area Network.
LoS	Line of Sight.
LPWAN	Low Power Wide Area Network.
LSTM	Long Short-Term Memory.
LTE	Long-Term Evolution.

LTE-M	Long-Term Evolution Machine Type Communication.
MCL	Maximum Coupling Loss.
ML	Machine Learning.
MME	Mobility Management Entity.
MNO	Mobile Network Operators.
MO	Mobile Originated.
MQTT	Message Queuing Telemetry Transport.
MT	Mobile Terminated.
MTC	Machine Type Communication.
Multi-RAT	Multiple Radio Access Technology.
NAT	Network Address Translator.
NB-IoT	NarrowBand-Internet of Things.
NIS	Network and Information systems.
NLoS	Non Line of Sight.
OTDOA	Observed Time Difference Of Arrival.
P-GW	Packet Data Gateway.
PCB	Printed Circuit Board.
PD	Participatory Design.
PD	Packet Drop.
PDP	Packet Data Protocol.
PIN	Personal Identification Number.
PPUSA	Periodic Uplink Scheduling Algorithm.
PRB	Physical Resource Block.
ProSe	Proximity Service.
PSM	Power Saving Mode.
QAM	Quadrature amplitude modulation.
QoL	Quality of Life.
QoS	Quality of Service.
QPSK	Quadrature Phase Shift Keying.
RAN	Radio Access Network.
RAT	Radio Access Technology.

RCT	Randomise Control Trial.
REST	Representational State Transfer.
RF	Random Forest.
RSRP	Reference Signal Received Power.
RSRQ	Reference Signal Received Quality.
RSSI	Received Signal Strength Indicator.
RTT	Round Trip Time.
S-GW	Serving Gateway.
SCEF	Service Capability Exposure Function.
SF	Spreading Factor.
SIM	Subscriber Identity Module.
SNR	Signal-to-Noise Ratio.
SPI	Serial Peripheral Interface.
SVM	Support Vector Machine.
TCP	Transmission Control Protocol.
TETRA	Terrestrial Trunked Radio.
TLS	Transport Layer Security.
TTRN	Transatlantic Telehealth Research Network.
UART	Universal Asynchronous Receiver/Transmitter.
UDP	User Datagram Protocol.
UE	User Equipment.
UI	User Interface.
UL	UPLink.
UUID	Universal Unique Identifier.
UX	User Experience.
VoLTE	Voice over Long-Term Evolution.

1 Introduction

The Internet of Things (IoT) has experienced rapid and exponential growth in the past few years and has shown no sign of slowing down. IoT is a network of connected devices that can sense and interact with the surrounding physical environment and communicate using the network, facilitating the seamless transfer of information. Figure 1.1 shows this growing trend of IoT devices over 15 years. According to Ericsson, by 2028, the total number of connected devices is expected to grow to 34.7 billion from just 17.7 billion in 2023. Cellular Internet of Things (C-IoT) alone will contribute to about 17.3% of total devices, about 6 billion [1]. The applications of the IoT have expanded over many sectors of business and society, including consumer electronics, agriculture, utilities, transportation, healthcare, etc.

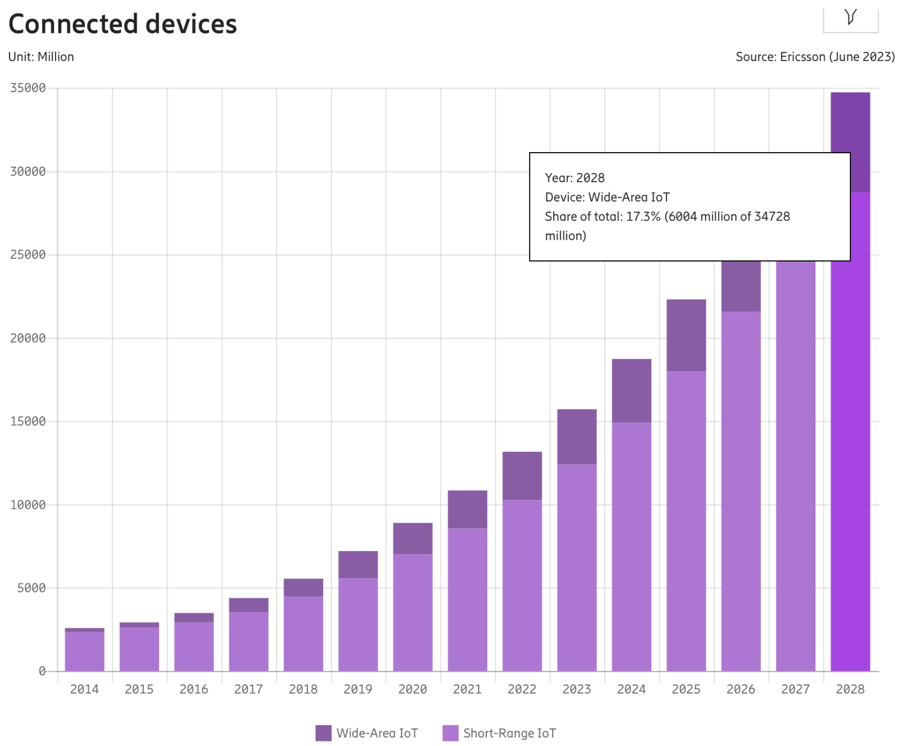


Figure 1.1: Ericsson Internet of Things (IoT) Devices Forecast

[2]

Several reasons contribute to this growth of IoT devices, the primary one being the availability of different network technologies developed specifically for IoT applications (such as LPWAN, NB-IoT, etc.) and the second being the affordability of hardware technologies to build IoT applications (LoRaWAN module from Microchip for 11 EUR [3]). IoT can improve different industries by improving efficiency, providing valuable insights, and automating processes.

The primary research question addressed in this dissertation for the technical study of this project was:

To verify the use of publicly deployed network infrastructure of the C-IoT communication technologies are sufficient to meet the specific requirements of critical IoT application or *is there a need for a parallel infrastructure to support the stringent demands of the critical applications.*

In order to perform the pragmatic evaluation of C-IoT, it was decided first to develop a critical healthcare application using the C-IoT technologies and use the expected performance requirement of such an application to evaluate the performance of the C-IoT technologies.

Several reasons make deploying a realtime critical application over an IoT network challenging. The LPWAN IoT technologies are spread across licensed and unlicensed spectrum of frequencies. It is challenging to build a realtime application over unlicensed LPWAN IoT alone due to duty cycle restrictions imposed on them. In Denmark, and the rest of Scandinavia, the C-IoT technologies are deployed primarily in the low-band (Band 20 in most Scandinavian countries). According to a report from Global System for Mobile communications Association (GSMA), low-band frequencies are now being used worldwide by Mobile Network Operators (MNO) for 5G deployment. This helps extend the 5G coverage in rural areas and decrease deployment costs. This makes it challenging for continuous monitoring IoT applications to have enough radio resources for uninterrupted operations. In busy hours, the DownLink (DL) speeds offered by low-band are up to 81% lower than that of mid-band frequencies [4].

The Quality of Life (QoL) of patients with chronic illness such as Cardiovascular Disease (CVD) can be improved using IoT technologies. CVD have the highest mortality rate worldwide, accounting for between 13% to 15% of all deaths worldwide. The number is even higher in Europe, contributing to about 24.8% of deaths due to CVD [5, 6]. Out of all the CVDs, Heart Failure (HF) is the most commonly diagnosed cardiovascular disease, with just over 37 million of the world's population being diagnosed with heart failure [7]. HF is causing a growing burden on the healthcare sector due to poor prognosis, unhealthy lifestyle, obesity, and increasing prevalence. This, combined with the increasing aging population, has started to put a financial strain on healthcare systems worldwide [8].

Multiple studies worldwide have shown rehabilitation programs' benefits and effectiveness for chronic diseases [9]. In the case of HF, cardiac rehabilitation is crucial to ensure improved & speedy recovery, improved Quality of Life (QoL), and overall well-being. Cardiac rehabilitation includes interventions like physical activity, weight control, improved diet, etc. These interventions are essential for a patient's recovery, but unfortunately, cardiac rehabilitation program suffers from poor compliance and adherence. Patient's have been compliant with taking the medication regularly but have failed to comply with the lifestyle changes [10]. In order to simplify and increase adherence to the rehabilitation programs, they started using more technology and telerehabilitation during the process. Traditionally, the telerehabilitation program has focused on collecting the data from patients and transferring it to medical professionals who can determine the improvements and worsening of HF. In most cases, the patients were uninvolved and exposed to their health data. Multiple studies highlighted that telerehabilitation was more effective when both the patients and the medical professionals were given access to the health data [11, 12, 6, 13].

This application developed in this PhD project is based on the Future Patient Telerehabilitation (FPT) project and leverages research outcomes from the project [14]. The focus behind FPT is the development of a telerehabilitation program designed to improve patient's QoL and empower them to perform self-monitoring to detect the worsening of their health, which can avoid re-hospitalization. In multiple interactions with medical doctors and researchers, it was understood that Electrocardiogram (ECG) could help detect and classify different CVD, such as arrhythmias, coronary artery diseases, heart failure, cardiomyopathies, etc. Some medical professionals believe that using on-demand or semi-realtime access to patient's ECG data can help detect some CVD early [5, 15, 16].

In order to increase the adaptation process, the project followed the Participatory Design (PD) methodology, which aimed to include all the stakeholders, e.g., patients, doctors, medical professionals, researchers, etc., while designing a continuous ECG and HR monitoring system [17, 18]. Multiple studies have highlighted that the eHealth system's success depends on the end-user's inclusion in the design process. Several other benefits of Participatory Design include improved user satisfaction, increased user engagement, reduced development time and cost, etc. [19, 20].

Figure 1.2 shows the different phases of the project and Participatory Design phases. As can be seen, the overall Participatory Design process was divided into three phases, each focusing on different aims, data collection techniques, and timelines.

1. **Phase I:** The first phase of the Participatory Design process focused on three aims.

    (a) The first was to understand the end user's perspective regarding continuous ECG and HR monitoring applications, their psychological and technical readiness levels, and their expectations.

    (b) The second aim was to address the pain points of the medical professionals and the challenges that can be considered while building such a system.

    (c) The third aim was to validate the initial prototypes and conceptual understanding of the problem that this system aims to solve.

    The first phase included different data collection techniques; some involved visiting the patient's home in Viborg and Skive (cities in Denmark) to ask them about their experience with the disease, their thoughts on continuous ECG and HR monitoring, wearing ECG sensors and IoT devices for extended periods, etc. Understanding patient's views on technologies and devices they would like to use in a workshop in Viborg. Another group of users of this system were medical professionals like cardiac doctors, cardiac nurses, other medical professionals, etc. In order to capture their understanding, a workshop was arranged where they were presented with early ideas and drawings of a contineous ECG and HR monitoring system and were asked for feedback. The system also was presented in summer schools organized by Transatlantic Telehealth Research Network (TTRN) in 2019 and in a Japanese & Danish Research Network on Telehealth/Telerehabilitation and Welfare Technologies (JD TeleTech). The timeline for this phase was between August 2019 to March 2020 A.

    Unfortunately, this project phase was short-lived due to the rise of the Covid-19 pandemic worldwide, which resulted in very limited access to cardiac patients and medical professionals. In order to move forward, the project decided to continue with the Participatory Design approach by involving researchers in the healthcare sector and students from diverse backgrounds.

2. **Phase II:** This project phase involved designing and implementing the IoT end device, web portal, and Android app based on the initial feedback received from phase I of the project.

    (a) The first aim of the phase was to evaluate the design principles and the prototype designs of the end device, web portal, and Android app.

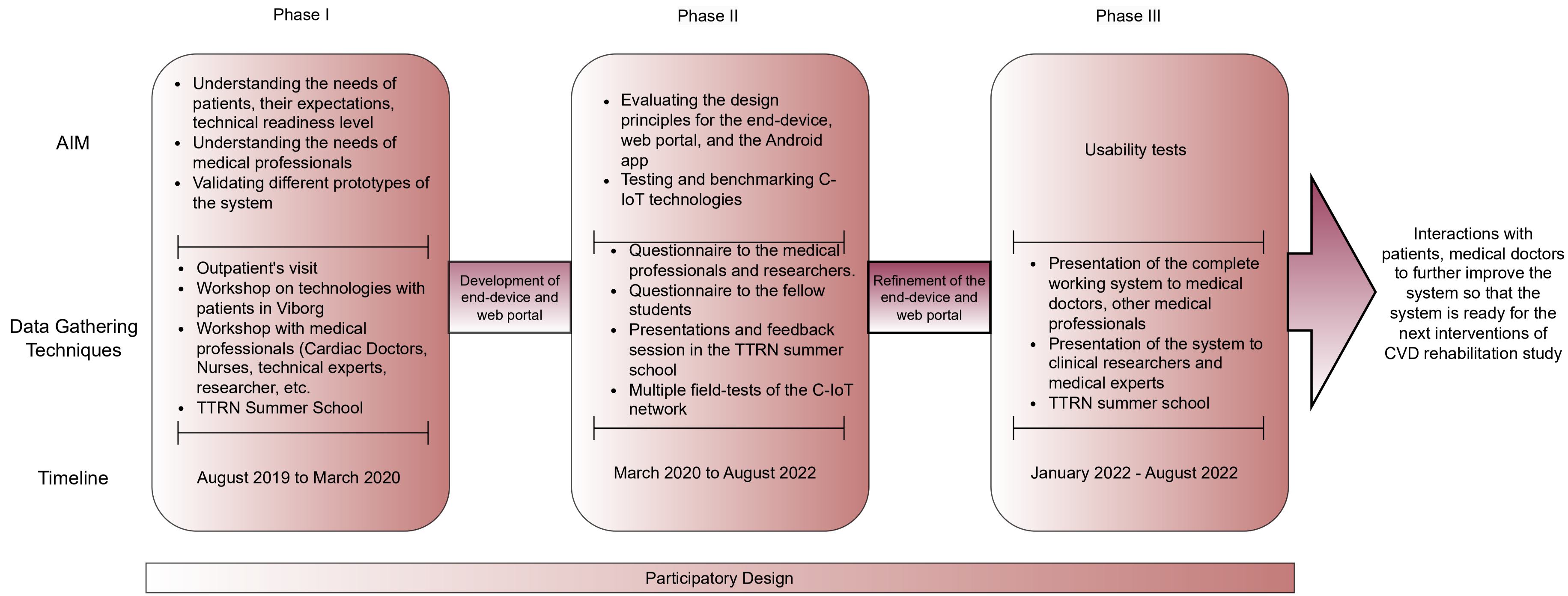


Figure 1.2: Different Phases of Participatory Design (inspired from [18])

(b) The second phase aimed to perform network tests to verify whether or not the designed application can meet the requirements identified in Phase I.

The data-gathering techniques used in this phase were divided into two. The first was a questionnaire sent out to medical professionals and domain experts to get feedback on the User Experience (UX) design and the usability of the web portal, and the second involved presenting the end device and the web portal virtually to medical researchers and fellow PhD students from Denmark, USA, and Japan in the TTRN summer school 2020. The primary goal behind the presentation at the summer school was to gather feedback that could be used to develop the system further. The timeline for this phase was between March 2020 to August 2022.

3. **Phase III:** All aspects of the system were further improved based on the feedback from Phase II of the project. Phase III, the final phase of this project, focused on performing several usability tests to identify potential challenges that can be solved using such a system. The data gathering technique used in this phase was to arrange workshops with the industry leaders, experts in the CVD field, and medical doctors and professionals from different backgrounds. This involved presenting the live functioning of the E2E system in front of medical doctors from Neuroscience in Denmark and Sweden. A similar demo of the system was presented in front of the researcher from Aalborg University to get more technical feedback on the overall system. The timeline for this phase was between January 2022 to August 2022.

The following section describes the high-level functional requirements set on the continuous Electrocardiogram (ECG) and HR monitoring system at the beginning of the project. The requirements are based on building a hypothesis based on the outcome from the different Participatory Design phases.

1. The ECG and HR monitoring should be performed in near realtime. This can allow medical professionals to look into the incoming ECG data to notice different patterns. The realtime data, if possible, should include both heart rate and ECG.

2. The monitoring device should not restrict the mobility of the individual wearing the device. The device mobility should not be restricted to indoor environments such as hospitals, homes, offices, etc.

3. The visible parts of the system should blend in for everyday wearing and should not stand out, indicating that the person wearing such a device is undergoing medical treatment.

4. The system should be able to provide continuous ECG and HR for extended periods without loss in data.

5. The monitoring device should serve as a generic platform for further project expansion. For example, the device should not be limited to only measuring heart activity but also support interfacing with other sensors to create an ecosystem of data points. In other words, the monitoring device should act as a gateway towards further project expansion, including different sensors.

6. The data collected by the monitoring device should be visualized and stored for later consumption. The visualization system should allow import from other systems and export data for further processing.

7. There should be a way for CVD patients to see their data to track their health.

8. There should be a possibility to develop Machine Learning (ML) algorithms that can assist medical professionals in understanding the patient's health condition.
9. The developed system should support on-demand as well as continuous monitoring of the patients.

The requirements above for building the continuous ECG and HR monitoring application helped us identify the approach to benchmark the performance of the C-IoT technologies:

1. Evaluating the performance and reliability of C-IoT in different coverage environments to carry critical application data in an E2E system.
2. Evaluating the performance of C-IoT technologies over an extended period of time.
3. Mobility and roaming support offered by C-IoT technologies.
4. Evaluating the scalability aspects of C-IoT technologies.

Overview of the Chapters

The content of this dissertation is divided into two parts. The first part of the dissertation focused on building an E2E continuous ECG and HR monitoring system for monitoring the heart activity of CVD patients. The second part of the thesis evaluates the KPI for C-IoT technologies by performing several experiments.

Chapter 2 provides an overview of different communication technologies used during this project. This includes an introduction and overview of the LPWAN technologies such as Long Range Wide Area Network (LoRaWAN), NarrowBand-Internet of Things (NB-IoT), and Long-Term Evolution Machine Type Communication (LTE-M).

Chapter 3 describes the development journey of the different prototypes of the end devices which were built during this project. The chapter also provides an overview of different ECG sensors discovered during the project and an overview of the development and validation of all prototypes developed during the project.

Chapter 4 provides an overview of the developed server and backend system. This includes the different prototype versions of the data collection & storage system and data visualization systems such as the heart rater, live web portal, and eMed Android app. The chapter also gives an insight into the different ML algorithms developed during this project.

Chapter 5 presents the results from several Key Performance Indicator (KPI) tests conducted using the C-IoT technologies, NB-IoT and LTE-M in multiple test environments. This includes performing the KPI tests of C-IoT technologies in indoor, deep indoor, outdoor, remote outdoor, and roaming conditions using multiple MNO. NB-IoT and LTE-M have also been tested for their power usage in different network conditions with different communication modules and per eNB device capacity.

Chapter 6 focuses on the experimental evaluation of different techniques that can help further improve the developed system's reliability. This includes an experimental implementation of Multiple Radio Access Technology (Multi-RAT) and Device-TO-Device (D2D) communication. The project develops a new prototype to support all the necessary technologies (including the fallback technologies) to send the data.

At last, Chapter 7 highlights the conclusion and key research findings and summarises the whole document. The chapter also provides an outlook on the possible future research directions.

2 Technical overview

2.1 Internet of Things (IoT)

The general term Internet of Things (IoT) can be defined as the network of connected devices that can sense and interact with the neighboring environment and communicate that information with the cloud. Wide area communication technologies for IoT have existed ever since the introduction of 2G in the early 90s. Since then, the number of new and advanced communication technologies and connected devices has increased. A typical IoT system consists of three main components: an end-device, a communication technology, and a cloud solution. Figure 2.1 shows a very generic implementation architecture of an IoT system.

- **End Device:** This is a device such as a temperature sensor, water meter, light meter, etc., that consists of a sensor to measure the surrounding environment, a processing unit to understand the analog or digital measurement from the sensor, a communication unit to exchange the information back-and-forth with the system, and an actuator to take action either based on measured values from the sensor or by decoding the instruction received from the cloud system.
- **Communication Technology:** The primary function of communication technology is to transfer measurement data and control signals between the end device to the cloud system. Depending upon the requirement of the application, a short-range or a long-range communication technology can be used. A communication technology designed explicitly for IoT applications often has two components a Gateway, which handles local wireless communication between the devices, and a long-range network technology that provides Internet access to the gateway. The choices of

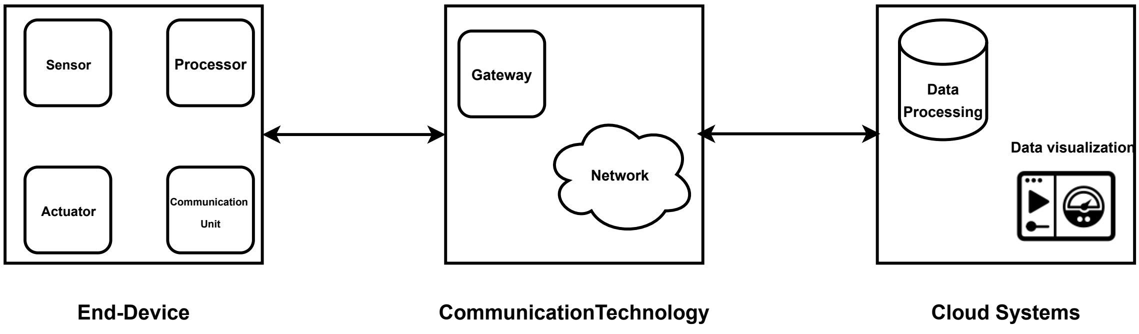


Figure 2.1: IoT generic implementation architecture

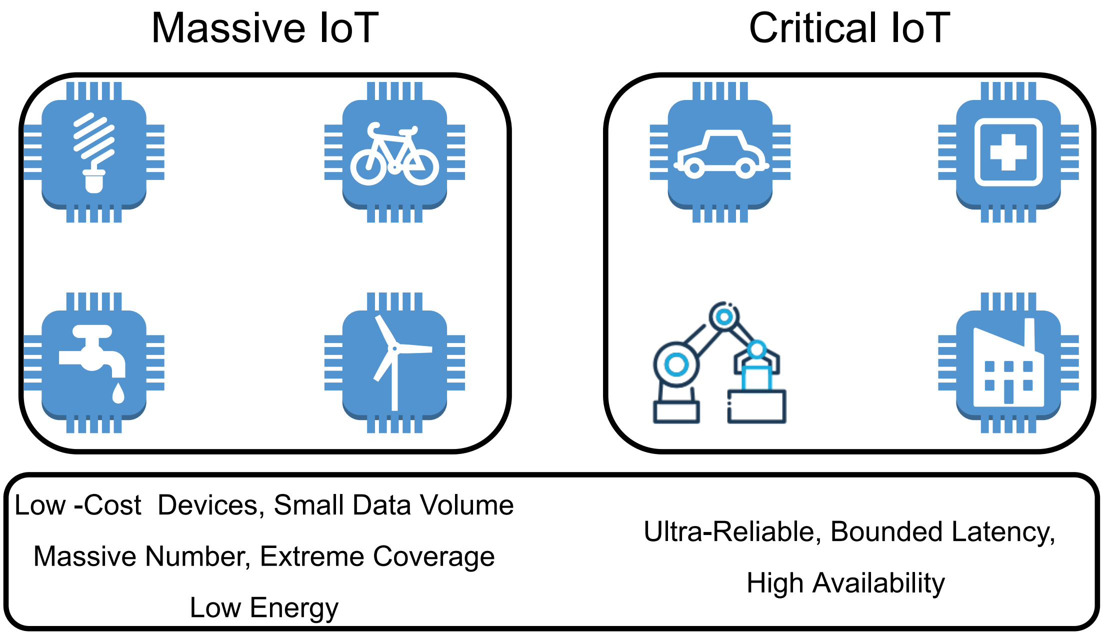


Figure 2.2: IoT application categories (inspired from [22])

these technologies are often dependent on but are not limited to the application requirements, user needs, and the available technologies in a given area.

- **Cloud System:** A cloud system in an IoT application is often a remote computing system or an online platform responsible for data processing, storage, and visualization. Cloud systems could have several algorithms, Machine Learning (ML) models for generating value, investigating patterns & trends, and providing deeper insights into the collected data.

IoT applications can be classified into different categories based on their characteristics, network requirements, and intended use. These categories include Massive IoT and Critical IoT. The performance requirements for these applications can vary in terms of mobility, security, reliability, latency, data rates, battery life, and cost of deployment. Cellular Internet of Things (C-IoT) technologies include 5G New Radio (5G NR), NarrowBand-Internet of Things (NB-IoT), and Long-Term Evolution Machine Type Communication (LTE-M), which opens up more categories of IoT applications such as broadband IoT and industrial automation IoT [21].

Figure 2.2 shows some examples of applications falling into different IoT application categories and their general characteristics.

2.1.1 IoT Connectivity Alternatives

For an IoT system to function properly, it requires connectivity or communication technologies. Depending on the specific IoT application, different communication technologies are utilized to provide access to these devices. To simplify things, we can classify these technologies based on the range of communication they offer.

When IoT applications are deployed in limited areas such as houses or office buildings, short-range communication technologies like WiFi, Bluetooth, Bluetooth Low Energy (BLE), and ZigBee are often used. These technologies operate in an unlicensed frequency spectrum, have limited battery life, varying data rates, and Quality of Service (QoS).

For applications that require long-range and wide area connectivity, there are two categories of connectivity options to choose from.

- **Cellular technologies:** This includes technologies introduced by 3rd Generation Partnership Project (3GPP), such as Second-Generation Cellular Network (2G), Third-Generation Cellular Network (3G), Long-Term Evolution (LTE), NB-IoT, LTE-M, 5G NR, etc. These technologies are primarily deployed and operate in the licensed spectrum. These technologies can provide Long-Distance coverage while simultaneously improving QoS, high data rates, lower interference, etc. Technologies like NB-IoT and LTE-M are specifically designed for IoT applications that demand extended coverage, long battery life, and reliable data transfer, among other things.

- **Unlicensed technologies:** Due to the demand for IoT and the wide range of applications it covers, several other proprietary solutions have been developed to satisfy the needs of Long-Range IoT applications. Sigfox and LoRaWAN are examples of such technologies requiring a dedicated end-to-end infrastructure. These technologies often work in ISM bands which have some limitations when using radio resources. Sigfox and LoRaWAN must follow duty-cycle restrictions (transmitter power, transmission interval, etc.) while transferring the information from the end device to the rest of the infrastructure.

During this project, several of these short-range and long-range technologies were tested and evaluated. Section 2.2 describes in detail the state of the art of some of the communication technologies which were used during the project.

2.2 Overview of Low Power Wide Area Network (LPWAN)

As the name suggests, the Low Power Wide Area Network (LPWAN) is a wireless technology that connects long-distance, battery-powered, low-bit-rate IoT applications. When it comes to the deployment of wireless IoT applications, there is a wide range of technologies that can be used. Each of these technologies varies in their offerings in terms of link budget, power consumption, range, bitrate, etc. Figure 2.3 shows the overview of different communication technologies typically used when interconnecting and deploying IoT applications.

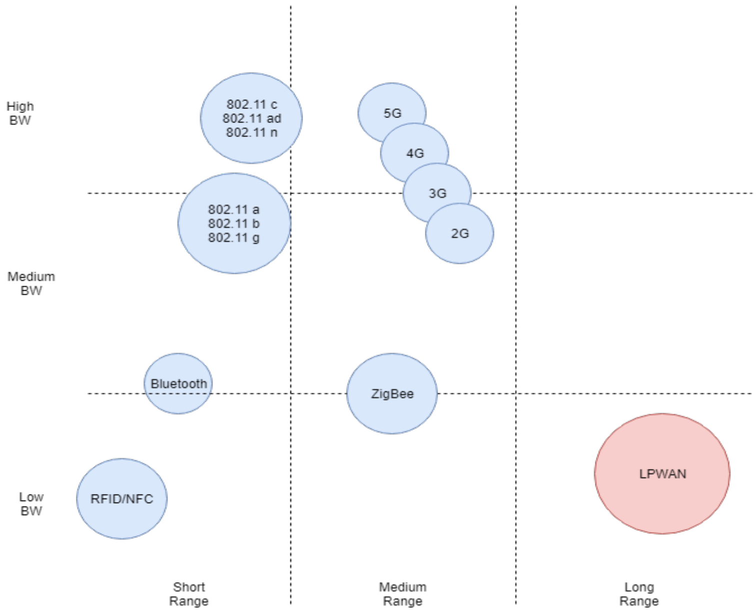


Figure 2.3: Wireless Technologies for IoT [23]

LPWAN technologies support some general characteristics,

- Long deployment rage of upto 20 km
- Low power consumption and support for longer battery life of upto 10 years
- Low data rate support upto 375 kbps
- Low subscription cost

LPWAN technologies can be divided into two main categories: Cellular Internet of Things (C-IoT) technologies and non-cellular IoT technologies. Cellular technologies, developed by 3GPP, aim to support LPWAN IoT development. NB-IoT and LTE-M are examples of C-IoT technologies. On the other hand, non-cellular IoT technologies such as Sigfox and Long Range Wide Area Network (LoRaWAN) are also commonly used.

This chapter will focus on the LPWAN technologies used in this PhD project, specifically LoRaWAN, NB-IoT, and LTE-M. These technologies will be described in detail in the following sections.

2.2.1 Long Range Wide Area Network (LoRaWAN)

Long Range Wide Area Network (LoRaWAN) is a wireless network protocol and system architecture that uses LoRa as a physical layer wireless communication link. This allows end-devices to send data to an application server. The standard is developed and operated by LoRa Alliance [24]. LoRa is a wireless modulation technique that has a high tolerance to interference. This allows the LoRaWAN network to provide robust coverage with low energy consumption. Chirp Spread Spectrum (CSS) is used for communication between the end node and the gateway due to its robustness to interference, long communication distance, and bi-directional communication. CSS has been used in military applications for decades (e.g. tracking, encrypted communication for unmanned ground vehicles, etc. [25, 26]) .

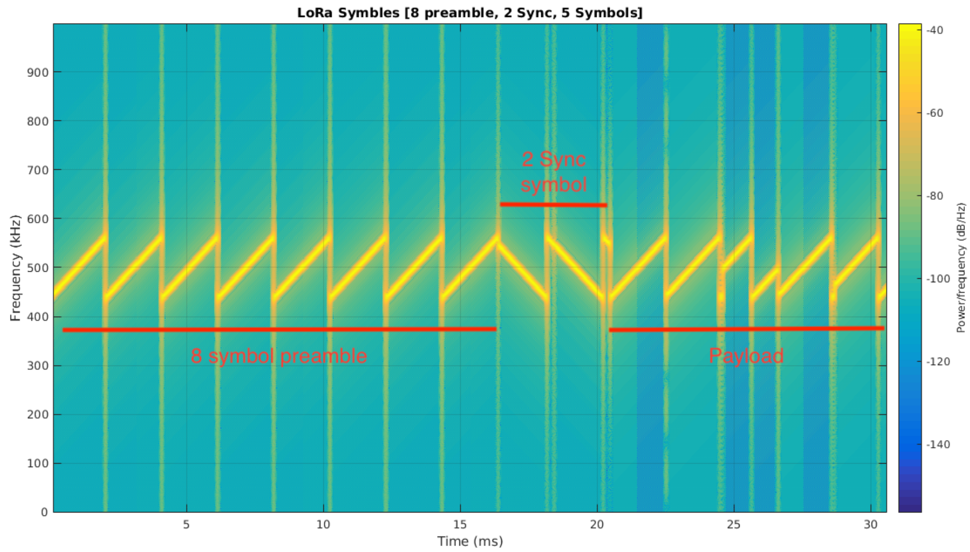


Figure 2.4: Example of LoRaWAN CSS [27]

Figure 2.4 shows an example of LoRa CSS modulation. In this example, it can be observed that the first up chirps are 8 preamble symbols followed by 2 down chirps which are synchronisation symbols. After that the actual payload is transferred [27].

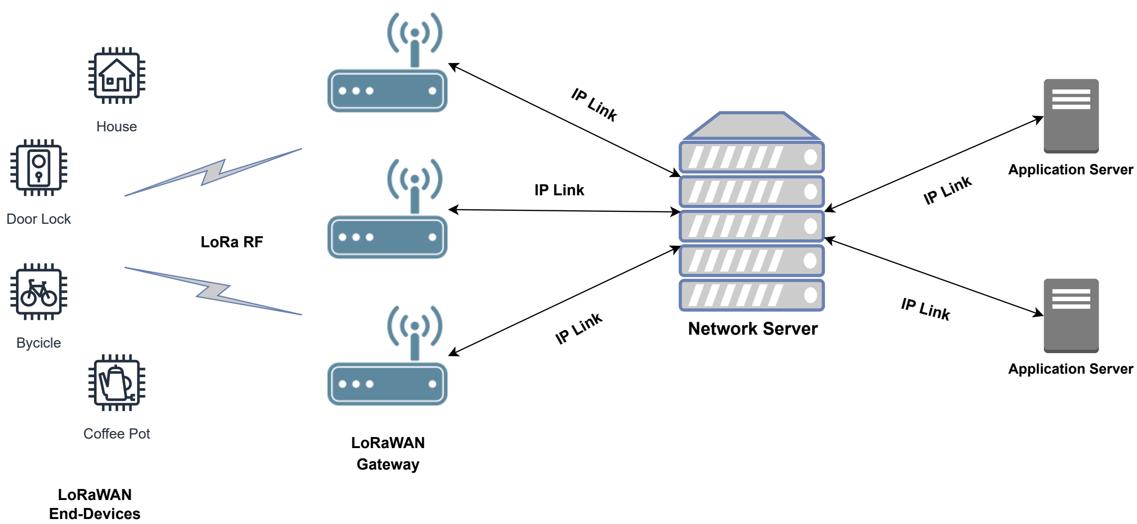


Figure 2.5: LoRaWAN Network Architecture (based on [28])

Most of the LoRaWAN deployed networks follow the star topology in the access network mainly to preserve the battery life of the end nodes. Figure 2.5 shows the E2E network architecture of LoRaWAN. LoRaWAN network can be deployed nationwide by one operator, but the private deployment of the technology is also possible. LoRaWAN primarily consists of the following network components:

1. LoRaWAN end devices: End devices are typically equipped with sensors, actuators, etc., with a LoRaWAN communication module attached. After collecting the sensor(s) data, these devices forward the information to the application server using

the LoRaWAN network. According to the LoRaWAN specification, the end devices fall into one of three classes Class A, Class B, and Class C.

- Class A devices support the lowest power consumption while simultaneously offering bi-directional communication. Class A device opens two receive windows after an uplink transmission to receive the downlink message from the gateway. The windows are opened after 1s and 2s, respectively. In the case of Class A devices, the downlink message from the server at any given time will have to wait until the next uplink message from the device.

- Class B devices also offer bi-directional communication and a scheduled timeslot for any downlink communication. Class B devices, in a way, are an extension of Class A devices. Class B devices use the gateway's time synchronization beacon frame to register for a fixed downlink slot. Class B devices are designed more for applications that send more downlink messages.

- Class C devices are designed to have maximum receive slots. Class C devices have no power restrictions and offer a continuously open receive window to receive downlink transmission. Class C devices use a similar method as class A devices for uplink transmission during which the receiver window is closed. However, as soon as the uplink transmission is finished, the device opens the receive window. Class C devices have higher power consumption than Class A and Class B devices but, at the same time lowest latency for communication between the server and the end device.

2. LoRaWAN Gateway: The basic function of a LoRaWAN gateway is to listen to all the incoming messages on all available LoRa frequencies. Once the gateway has received a packet from a LoRaWAN device, it adds metadata to the packet. The metadata could contain timestamps, frequency spreading factor, gateway id, Received Signal Strength Indicator (RSSI), etc. After this, the data is forwarded to the network server using an IP link. Unlike many other wireless communication technologies, in LoRaWAN, the devices are not attached to the gateway. Instead, once the device is registered on LoRaWAN, all the gateways belonging to the same LoRaWAN network and within the range can receive the message from the LoRaWAN device. Once the message is received, all the gateways add the metadata and send it to the network server.

3. LoRaWAN Network Server: In the case of LoRaWAN, the network server is where all the complex functionality is stored. Some of the primary tasks carried out by the network server are to authenticate the device, accept all the incoming packets (including the duplicate packets), store metadata, forward device payload to an application server (if specified), LoRaWAN V1.1 also supports location service using multi-lateration techniques, send downlink messages to the device, etc..

4. Application Server: The application server is usually not part of the LoRaWAN infrastructure and is mainly set up by individuals to receive data from the LoRaWAN devices. Application servers can contain logic to understand the payload from the device and may contain different commands for the end device based on the received data. The application server can send actions to the end device by sending a LoRaWAN downlink frame to the network server, which then forwards that information to the gateways and then to the end device.

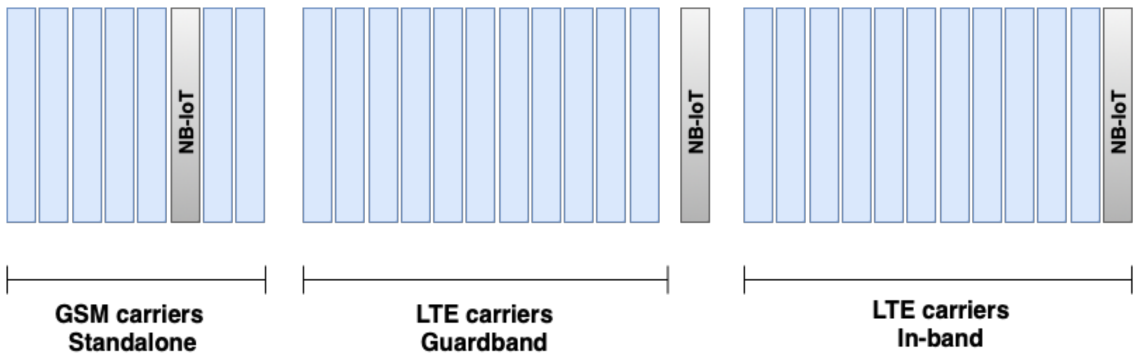


Figure 2.6: NB-IoT deployment modes (based on [34])

2.2.2 NarrowBand-Internet of Things (NB-IoT)

NarrowBand-Internet of Things (NB-IoT) is a cellular Machine-Type technology described first in the 3GPP release 13. NB-IoT standard shares many of its features on the existing LTE standard. NB-IoT, amongst other things, primarily focuses on providing deeper indoor coverage, massive device deployment, lower cost of deployment, and longer battery life. Release 13 of NB-IoT supports the Maximum Coupling Loss (MCL) of 164 dB, which is almost 20dB higher than the standard LTE, up to 52547 devices handled by a single cell. A higher MCL of 164dB translates into better deep-indoor coverage; this is achieved via the combination of newly introduced techniques of message repetition in both UL and DL messages and increased Power Spectral Density [29]. NB-IoT supports Extended Coverage (EC) classes, EC0, EC1 and EC2. NB-IoT supports a maximum of 2048 repetitions in the DL directions and 128 repetitions in the UL directions [30]. NB-IoT uses Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK) modulation schemes. This helps reduce the complexity of wireless communication but comes at the expense of reduced data rates. According to the NB-IoT standard described in release 13 of 3GPP, there is no support for voice or Voice over Long-Term Evolution (VoLTE) calls or IP Multimedia Subsystem (IMS) services. NB-IoT standard does not support the handover functionality in LTE, and there is no good choice of technology for applications requiring mobility support. In order to keep implementation less complex, NB-IoT only supports half-duplex communication, which means the NB-IoT UE can either transmit or receive data at a time from the eNB [31, 32].

NB-IoT uses a carrier of 180kHz bandwidth equal to one Physical Resource Block (PRB) of LTE. Depending upon where the 180kHz carrier is deployed NB-IoT standard has built-in support for three ways of deployment: In-Band, Guard Band, and Standalone [33].

Figure 2.6 shows the different deployment modes of NB-IoT.

- **Standalone** mode of deployment allows NB-IoT deployment on a single 200 kHz timeslot of GSM with a 10 kHz guard band on either side.
- **In-band** deployment mode utilizes one PRB from LTE resources inside the LTE band.
- **Guard band**, as the name suggests, allows NB-IoT to be deployed in the guard band around the LTE carrier.

NB-IoT supports Extended Discontinuous Reception (eDRX) and Power Saving Mode (PSM) to extend the device's battery life by optimizing power consumption. eDRX mode allows for extended timer values while listening to a DL message from the Evolved Node

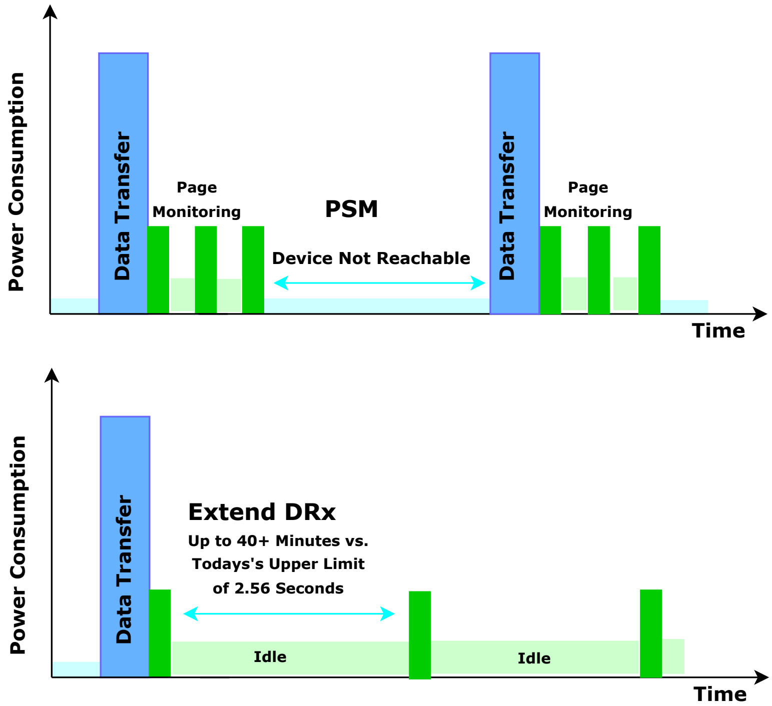


Figure 2.7: C-IoT power saving features (based on [35])

B (eNB). eDRX can be configured to a maximum value of three hours. In PSM, the device enters a low-power state to reduce unnecessary network signaling activity. This is ensured by maintaining the context of the device and at the network end. This optimizes the signaling required to change the state from idle to active mode. In addition to PSM and eDRX mode, NB-IoT also adapts to the Connect mode Discontinuous Reception (cDRX) with the maximum cycle value of 10.24s in 3GPP release 13. cDRX typically consists of a combination of a short and long DRX cycle. Combining all these advanced features for NB-IoT allows the technology to extend the device's battery life significantly [36, 37, 38]. Figure 2.7 an overview of different power savings mode.

Figure 2.8 give an overview of the data architecture of NB-IoT. Unlike traditional LTE systems, NB-IoT is designed to use both the control and data plane of the communication to transfer IP and non-IP messages. This is possible because of the optimization of the User Plane C-IoT Evolved Packet System (EPS) and Control Plane EPS.

1. Control plane optimization: The uplink data is transferred from eNB to Mobility Management Entity (MME), and the data is further forwarded based on IP or non-IP type. If the data traffic is IP, it is forwarded to the Serving Gateway (S-GW) and then to the Packet Data Gateway (P-GW). If the traffic is non-IP, it is forwarded to Service Capability Exposure Function (SCEF). After this, the data is finally forwarded to an application server. SCEF is a new node designed for IoT data, which helps deliver non-IP data from the control plane over the abstract interface.

2. User Plane optimization: In this case, both the IP and non-IP data follow the usual path. From the device, the uplink data reached the eNB. After eNB the data goes over S-GW to P-GW to the application server [32].

Subsequent releases form 3GPP for NB-IoT include improvements on bitrate (UL: 159 Kbps DL:127 Kbps), supporting lower power class devices (with 14 dBm), support for lo-

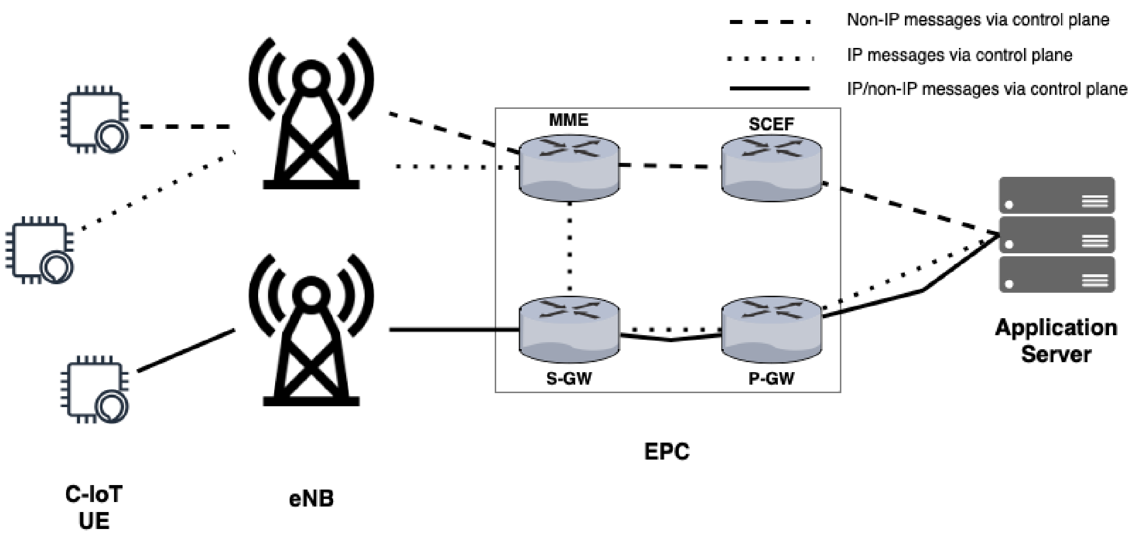


Figure 2.8: NB-IoT data architecture(based on [39])

calization (E-CID and OTDOA), wake up signaling, and support for small cell [32]. 3GPP has described improvements to the NB-IoT until release R16. The NB-IoT network evaluated during this project was deployed in 3GPP release R13.

2.2.3 Long-Term Evolution Machine Type Communication (LTE-M)

Long-Term Evolution Machine Type Communication (LTE-M) was introduced in 3GPP release 13, alongside NB-IoT. NB-IoT and LTE-M share many designs but can support different use cases. According to 3GPP release 13, LTE-M can support use cases with expected higher data rate, device mobility, lower latency, etc [33, 40].

Although NB-IoT and LTE-M are very similar by design, some differences allow them to complement each other, and combined technologies can support a much wider range of applications. Unlike in NB-IoT, LTE-M usually can only be deployed in-band and can occupy up to 6 Physical Resource Block (PRB) making a total of 1.4MHz bandwidth. LTE-M supports higher peak bitrates of 1 Mbps (both UL and DL) than that of NB-IoT. LTE-M supports more functionality, similar to LTE including device mobility, VoLTE calls, location based on Observed Time Difference Of Arrival (OTDOA), etc. Unlike NB-IoT, LTE-M also supports full-duplex communication. Higher bandwidth supported by the LTE-M standard makes it possible to use higher modulation scheme of 16 Quadrature amplitude modulation (QAM) [33, 40].

Like NB-IoT, LTE-M supports deep-indoor coverage more than LTE (at least by 20 dB). This is possible due to the support for repetitions in the standard. In case of LTE-M, the standard described support for two EC modes, mode A supports 32 repetitions whereas mode B supports 2048 repetitions. Mode A is the default mode of deployment for LTE-M, it is designed to support high bitrate IoT applications such as voice calls, mobility etc. Mode B is primaraly focuses on low-speed or stationary IoT applications where the data rates are not that important . LTE-M also supports longer battery life due to the support for features like PSM and eDRX mode. Like NB-IoT, the LTE-M technology supports IP and non-IP traffic over control and user plane[32, 37, 41].

Feature	LoRaWAN	NB-IoT	LTE-M
Bandwidth	125 kHz	180 kHz	1.4MHz
Modulation	Chirp spread spectrum	BPSK, QPSK	QPSK, 16 QAM
Bitrates	0.3 - 27 kbps	127 kbps (DL) 158 kbps (UL)	1 Mbps for DL & UL
Localisation	Yes	No (Rel. 13)	Yes
Voice Services	No	No	Yes (VoLTE)
Duplex Mode	Half-duplex	Half-duplex	Half-duplex, Full-duplex
Maximum Coupling Loss	NA	164 db	164 dB

Table 2.1: LPWAN feature comparison [33, 40]

Table 2.1 summarises some of the features of LoRaWAN, NB-IoT, and LTE-M discussed in the chapter.

After considering different features offered by all the LPWAN technologies, it was decided to continue with LTE-M as the primary communication technology for designing the continuous ECG and HR monitoring application. Following reasons were the primary factors contributing in making this decision:

1. LTE-M provides deep indoor coverage (at least 20 dB improvement over the traditional LTE standards).
2. LTE-M supports higher bandwidth as well as data rates when compared with data rates offered NB-IoT.
3. LTE-M supports possibility of initiating Mobile Originated (MO) and Mobile Terminated (MT) VoLTE calls.
4. LTE-M supports mobility, which is not supported by 3GPP release 13 for NB-IoT.
5. LTE-M supports power saving features such as PSM and eDRX.

3 End-Device

3.1 Overview self-monitoring devices

The use of health apps and wearable devices is increasing exponentially day by day , and people are becoming more conscious of monitoring their health. The global fitness monitoring devices market size is expected to jump from 30.41 billion USD to 114.36 billion USD by 2028 [42]. Self-monitoring technologies include wearable and non-wearable sensors, which are often integrated with a smartphone app or web interface. This allows individuals to collect and reflect on their health data. Fitbit tracker, Apple Watch, Samsung Galaxy Watch, blood-pressure monitors, sleep sensors, etc., are some examples of wearable monitoring devices. The baseline idea of these technologies is to enhance the Quality of Life (QoL) of the users by empowering and motivating them to take active participation in improving their overall health [43].

However, most of these self monitoring technologies are not designed with an understanding of the adequate needs of people with home-monitoring requirements. Another challenge with using these technologies is the lack of publicly available reports about clinical validation of these technologies, making it harder for medical professionals to rely on the data entirely [44]. Several studies have shown potential in supporting a patient's transition from the hospital back to home [45, 46, 47]. There are still some open challenges in doing this transition, namely technological acceptance by the patients and the family, with everyday life how-patients can install, transport, and use these monitoring devices to obtain the best possible results, lack of privacy measures, stigmatization (due to carrying visible sensors, etc.), reliability of the Information and Communications Technology (ICT) infrastructure for transportation of data from patient's home to the hospital systems [43].

Most remote monitoring sensors/devices, like Fitbit, Apple Watch, etc., collect user data and send it to a proprietary cloud system, making it harder for the medical professional to access it directly. Other monitoring devices like Electrocardiogram (ECG) holter maintains a local recording of the data for several days, which is then sent to the hospital by post for further examination. In both cases, performing an accurate and timely evaluation of the patient's health condition is tedious and time-consuming. The alternative is to use continuous monitoring of these critical health parameters and provide preliminary health analysis.

Figure 3.1: AliveCor KardiaMobile [50]

There are several popular at home ECG monitoring devices that are used by patients, researchers, and even clinical professionals. We decided to study the design of these devices. The two most commonly used devices recommend by the telemedicine researcher were AliveCor KardiaMobile and Cortrium [48, 49] .

Figure 3.1 and 3.2 shows the two devices developed by AliveCor and Cortrium. As can be seen from the diagram, both of devices use a different way of recording the ECG.

Working of AliveCor KardiaMobile

KardiaMobile provides a single lead ECG measurements that can be used for detecting atrial fibrillation, bradycardia, and tachycardia. The patient need to place the smartphone app running while performing the ECG measurement as can be seen from the figure 3.1. The deviced used Bluetooth to share the data with the smartphone app. Once the ECG is recorded, the patient needs to download and send the recording to the doctor. Although KardiaMobile had many good, clinically validated fetures, the biggest challenge in using it in the context of this project was the lack of ability to provide continuous recording of ECG and HR. KardiaMobile, in one measurement could only record the data for only 30 seconds which made is mismatch for the application targeted by this project [48].

Working of Cortrium

Cortrium C3 + holter monitor was a much better match in comparison to the KardiaMobile, this was mainly because of its ability of performing continuous ECG recording. The device also had a long battery life (up to 7 days) which made it more fit for the project. Cortrium out of the box is designed to continuously record the ECG and share the data with the smartphone app using Bluetooth. Cortrium provides a 3 channel ECG recording and also allows for the realtime visualization of ECG data in their smartphone app. The data sent by the Cortrium C3 + holter monitor uses a proprietary data structure that is not publicly available. Unfortunately our collaboration attempts with Cortrium were not successful

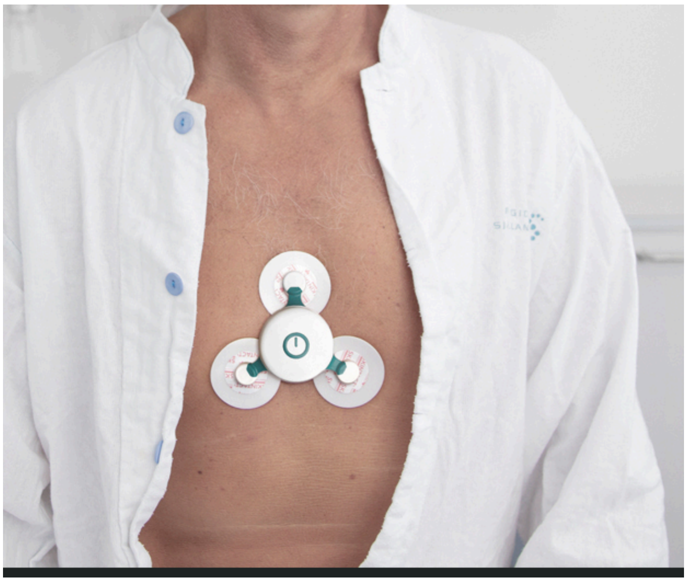


Figure 3.2: Cortrium Device: C3 + holter monitor[49]

therefore, even though C3 + holter monitor was really good match for the project, the use of Cortrium device had to be discarded.

3.2 Considerations for the end device

After initial discussions with the clinical researchers, medical professionals, and previous research studies, a development direction for the end device was established. Following were the design goals established at the beginning of the Phase I of the Participatory Design

1. The device should have an integrated ECG and HR monitoring sensor.
2. The device should have realtime data and control signal transmission capabilities.
3. The device should have a smaller footprint and be able to run on battery power.
4. The device should support user mobility, i.e., a person wearing the device should not be confined to one place (e.g., home, hospital, etc.) but should be able to move in-between environments.
5. The device should support a generic interface for data transfer between the sensor and the communication module in order to have easy expansion and integration of the future body sensors.

Figure 3.3 illustrates the conceptual design of the end device. Multiple prototypes of the described end device were designed during the project's duration. Multiple body sensors were identified and integrated with the end device during this time. Section 3.3 provides an overview of the different sensor and end device designs explored during the implementation of the project.

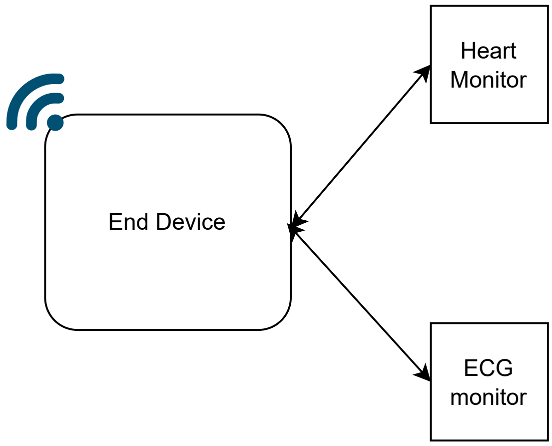


Figure 3.3: End Device Block Diagram

3.3 Sensor search

This section goes over the overview and the evaluation process of different ECG sensors.

Figure 3.4: AD8232 Sensor [51]

3.3.1 Single Lead Heart Rate Monitor: AD8232

AD8232 is a single lead HR monitoring sensor and allows for measuring the electrical activity within the heart. It can be used to measure the ECG of the person wearing the sensor and collect data into a microcontroller for further processing. Figure 3.4 shows the sensor and the three wired electrodes needed to gather the electrical signal from the heart. After receiving the data from the heart AD8232 combines the signals from all three electrodes and generates a single output ECG signal.

Figure 3.5 shows the first prototype developed during the course of the project. As can be seen from the figure, the prototype uses an Arduino UNO as a microcontroller connected

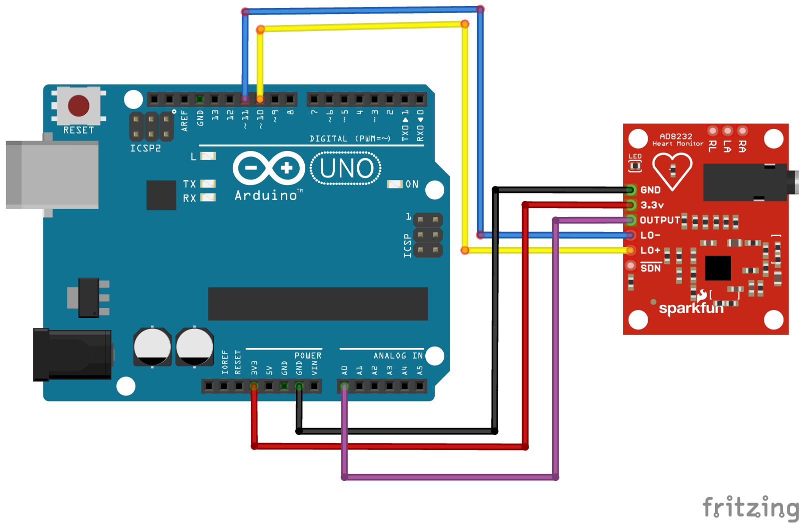


Figure 3.5: AD8232 prototype

to the AD8232. The two boards are connected using jumper wires, and output from the AD8232 is fed to the analog input pin of the Arduino UNO microcontroller.

There are several advantages of using an ECG device described in figure 3.5. The entire assembly of devices is cheaper than that of professional ECG equipment. Another advantage of this ECG device is that it is straightforward to set up and uses familiar electrodes for measurement.

The device in figure 3.5 was presented as a concept at an outpatient visit conducted in Skive, Denmark, and during the discussion, it was conclude that using a device like this is not preferred by the patients mainly for the following reasons:

1. The device is very bulky to carry around.
2. The device uses wires for connection; the patients gravitated more toward a wireless solution.
3. Patients prefer a plug-and-play solution rather than manually connecting different system components.

Apart from the issues discussed with the clinical users and patients, several technical issues were discovered with this implementation approach.

1. It takes much work to make a smaller footprint of the device.
2. The connector if loosely connected may introduce noise into the measurement due to the motion of the electrical wires.
3. There is no communication technology built into the device, making it very challenging to send the ECG data to a remote server without adding additional modules to the system. This meant connecting more wires to the main mcirocontroller board.

Therefore, due to all the issues described by the clinical users and considering technical challenges, it was decided to build a more compact solution with a special focus on including wireless technologies in the setup.

Figure 3.6: Movesense ECG sensor [54]

3.3.2 Wireless ECG sensors

There are several wireless ECG and HR monitors which are available. The selection of the sensor is primarily dependent on the following key criteria:

1. Programmability of the sensor, i.e., to choose the sampling frequency, measurement interval, etc.
2. Support for wireless communication technology, e.g., Bluetooth Low Energy (BLE), etc.
3. No use of wired electrodes to perform the measurement.

Based on the above criteria, we narrowed the search to two ECG and HR monitoring sensors Movesense [52] and Savvy [53]. Both of these sensors matched the required criteria set for the ECG and HR sensor. The following section goes over the technical details and the functionality offered by Movesense and Saavy sensors.

Movesense Medical:

Movesense is a BLE-based ECG and motion monitoring sensor which has undergone clinical Class IIa certification. Class IIa is a regulatory classification of medical devices in the EU. It indicates that the device follows specific standards and can be sold inside the EU. Class IIa devices are usually meant to be used in or on the human body for a limited period. Figure 3.6 shows the two different form factors of the Movesense ECG sensor. The Movesense ECG sensor has a smaller form factor, is comfortable, and is easy to put on. This can help patients use this sensor for extended periods without feeling fatigued. One of the key advantages of using this sensor is the wide range of programmability offered by the sensor; this enables the development of custom applications that are very specific for the use case. The sensor is also equipped with motion sensors, which can be further used to detect the motion of the patients alongside the heart activity. The sensor uses a coin cell battery and provides a long battery life which makes it easy to be used by people of all ages and categories [52, 54].

Savvy:

Saavy ECG was developed in collaboration with the Parallel and Distributed Systems Laboratory, Department of Communications Systems at Josef Stefan Institute. Saavy is

Figure 3.7: Savvy ECG sensor [53]

a CE-marked Class IIa medical device. Unlike Movesense, Saavy uses two electrodes that can be mounted at different locations to measure the ECG of the patient. Saavy use BLE to transfer the ECG data, and the sampling frequency can be customized. One of the important features of Saavy is that it comes with a rechargeable battery, which eliminates the need to replace batteries from the device. Saavy resembles the traditional portable ECG monitors, making them familer and easy to use. Figures 3.7 show the device and the app showing the ECG sample [53].

3.4 Making first prototype

In the last section, we narrowed down our search for the ECG and HR monitor to Movesense and Saavy. The primary reason for their selection was the use of BLE as a communication technology and the ability to program the sensors on the fly. In this section, we will focus on the first prototype, which was designed to test and verify the hypothesis of using a wireless ECG sensor to perform the ECG and HR monitoring of the patients. In order to make this prototype, we needed the following components:

- A Microcontroller with BLE transceiver
- Movesense and Savvy sensors
- Application to visualize the sampled ECG data

In order to make this prototype, ESP-32 was selected as the microcontroller. ESP-32 is a compact System-on-Chip (SoC) developed for low-cost and battery-driven applications. ESP-32 has a 32-bit CPU operating at 240MHz and can connect different peripherals using BLE, WiFi, UART, SPI, and I2C protocols. This enables us to not only connect both the ECG sensors to the ESP32 but also gives us the possibility to send the data to a remote database for storage. ESP-32 is one of the most popular microcontrollers amongst IoT developers because it has a good online support community and documentation available [55]. Figure 3.8 shows the ESP-32 development board used during the design of the first prototype.

The development approach was divided into two phases. Phase one focused on establishing a connection with the ECG sensors (both Movesense and Savvy), and phase two involved sending the data to a remote server for storage and visualization.

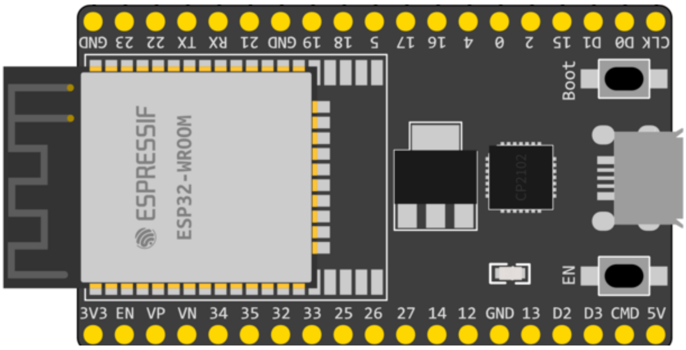


Figure 3.8: ESP-32 Development Board [56]

3.4.1 Integration with ECG sensor

Movesense ECG and Saavy both support BLE, but they had a different procedures to connect and start streaming the ECG and Heart. This section will go through the communication flow for both sensors.

Connecting Movesense with ESP-32

Figure 3.9 shows the flowchart for the BLE setup. Once the ESP-32 has finished booting, it activates the BLE radio and scans for the Movesense ECG sensor's Universal Unique Identifier (UUID). Movesense ECG allows modification of the UUID of the sensor in the firmware. In order to simplify the process, the UUID of the sensor was kept as the default value, and it was programmed in the ESP-32. The ESP-32 executes a periodic WHILE LOOP until the desired UUID is found. Once the UUID is found, the ESP-32 looks for the HR service; this service is also highlighted with its own UUID and also is programmed in the ESP-32. If the HR service is not found, the ESP-32 will print the error message on the serial monitor highlighting the reason for terminating the program. Once the HR service is found, the program looks for the notify characteristic of the service; this instructs the BLE module inside the Movesense ECG monitor to send a notification to the ESP-32 when the new ECG and HR data is available to fetch. If the notify characteristic is not found, the ESP-32 writes the error message onto the serial monitor. Once the correct characteristic is subscribed, the ESP-32 looks for a command characteristic that allows the program to feed data to the Movesense sensor. This data is nothing but a command constructed using different configuration values, which tells the sensor to start the ECG sampling and send that data back to the ESP-32. Once the correct command is fed to the Movesense sensor, the ESP-32 waits for new data to be received from the sensor. The received heart data is processed to extract ECG and HR from it and is plotted onto the serial monitor. Movesense ECG monitor keeps sending the new data until the BLE connection is active, or the manual termination of the data is requested.

Figure 3.10 shows the output data from the Movesense ECG sensor. The blue line indicates the ECG measurement whereas the red line indicates the HR of the person wearing the Movesense sensor.

Connecting Savvy with ESP-32

Figure 3.11 shows the BLE connection setup for Savvy ECG monitor. Most of the steps involved in establishing the connection between Savvy and Movesense are very similar,

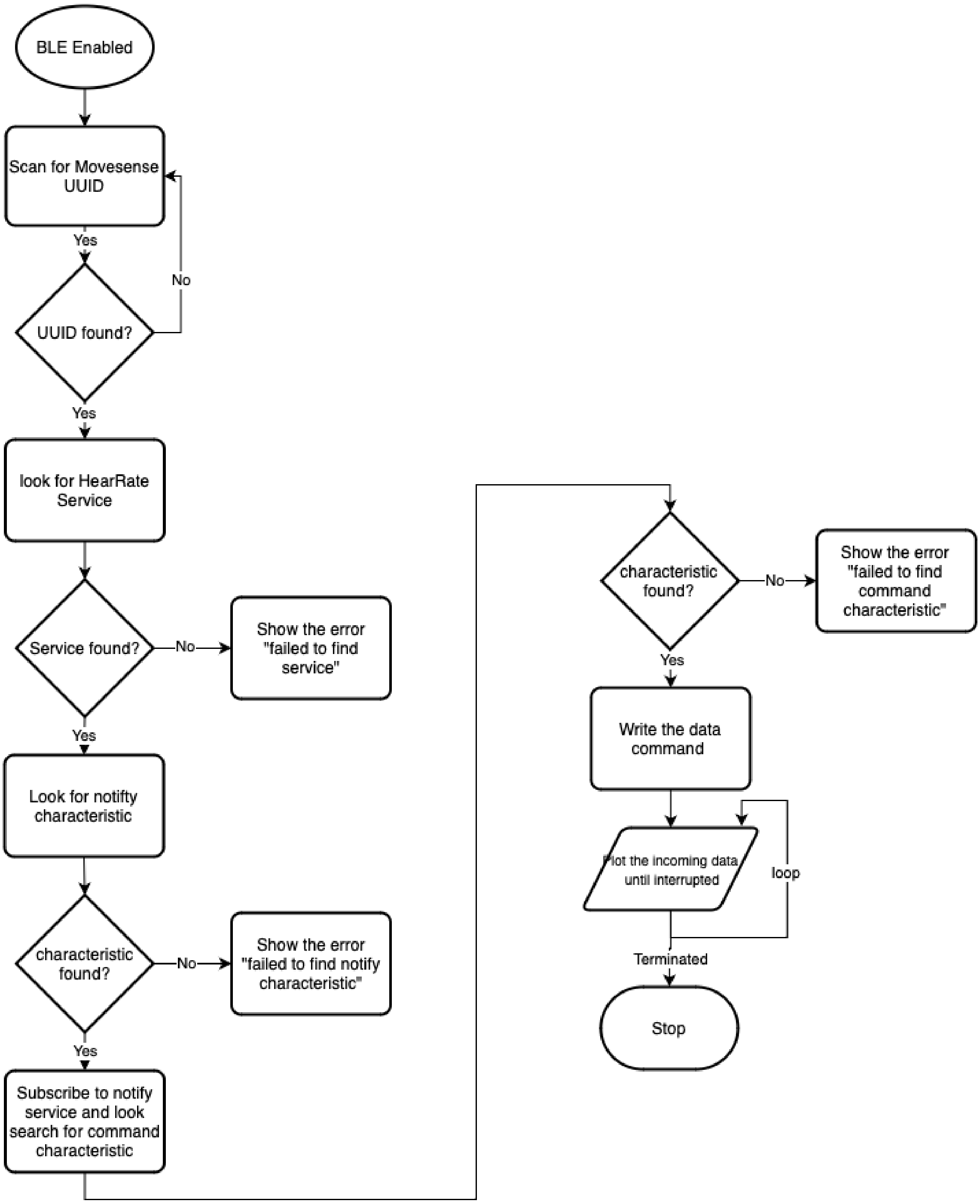


Figure 3.9: Movesense BLE Setup

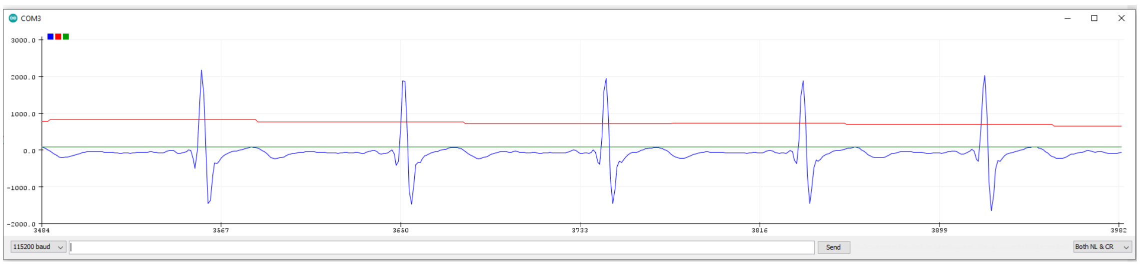


Figure 3.10: Movesense ECG output (Arduino IDE Serial Monitor)

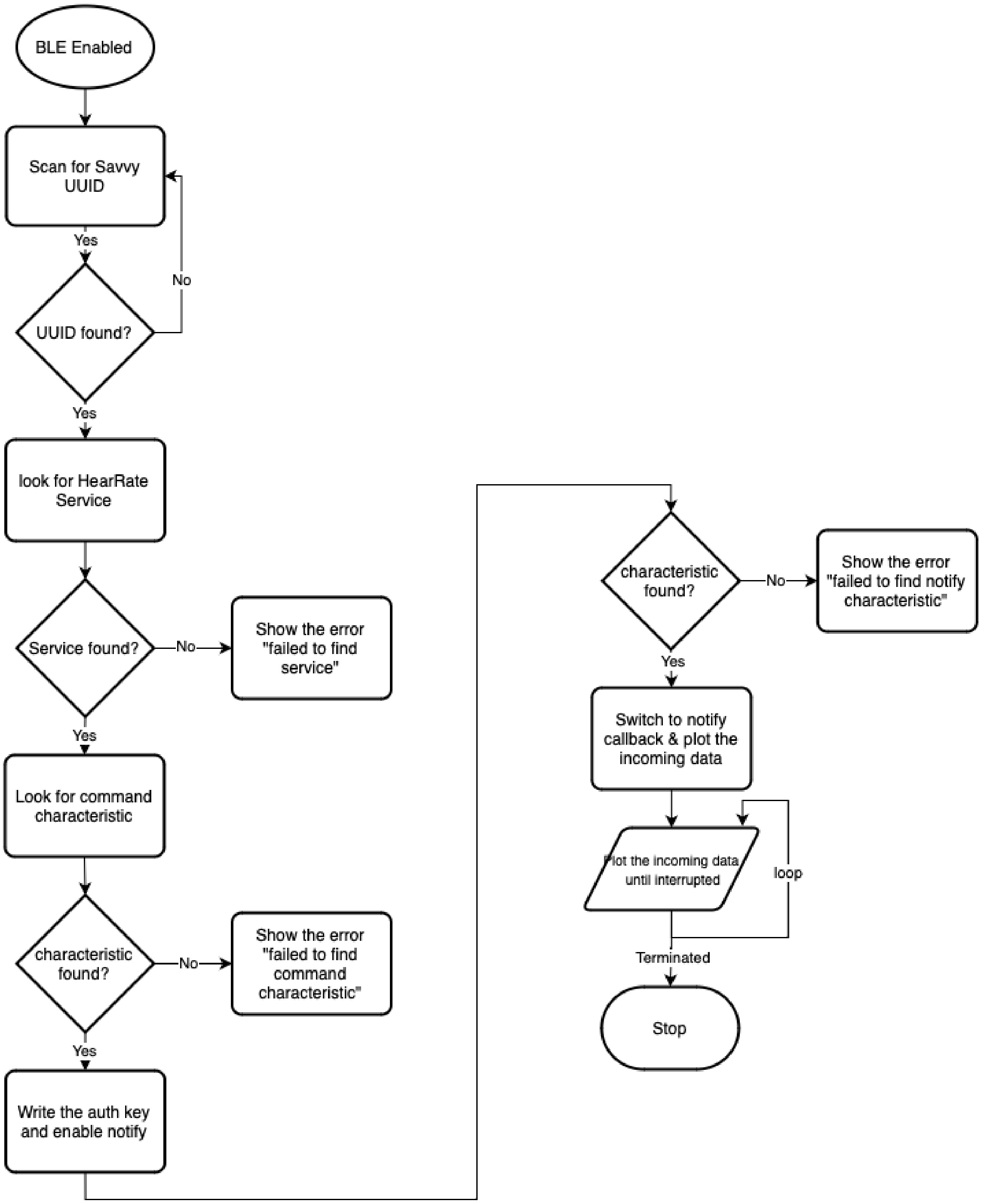


Figure 3.11: Savvy BLE setup

except for the data processing steps and the fact that the Savvy ECG monitor requires an authentication code. Like Movesense, while connecting Savvy ECG monitor, ESP-32 scans for the Savvy UUID, connects with the HR service, and searches for the command characteristics. In the case of Savvy, the sensor does not allow further communication unless the authentication procedure is completed within 4 seconds of the BLE connection. Once the ESP-32 has written the Personal Identification Number (PIN) code that will authenticate the user using the command characteristics, ESP-32 enables the notify characteristics. Savvy ECG starts sampling data immediately and notifying the ESP-32 with new data. ESP-32 processes this data, and the ECG and HR values are plotted onto the serial monitor. The command characteristics of Savvy can be used for further configuration Savvy, e.g., changing the sampling frequency, etc. The ECG data is continuously fed to ESP-32 until the BLE connection is broken or Savvy is asked to terminate the notify characteristics.

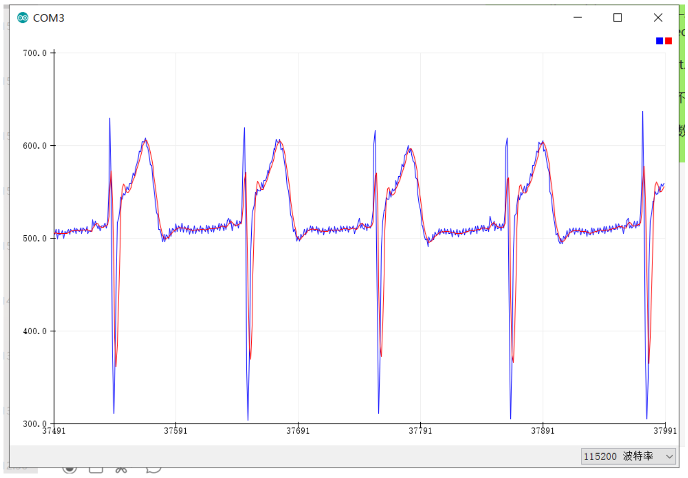


Figure 3.12: Savvy ECG output (Arduino IDE Serial Monitor)

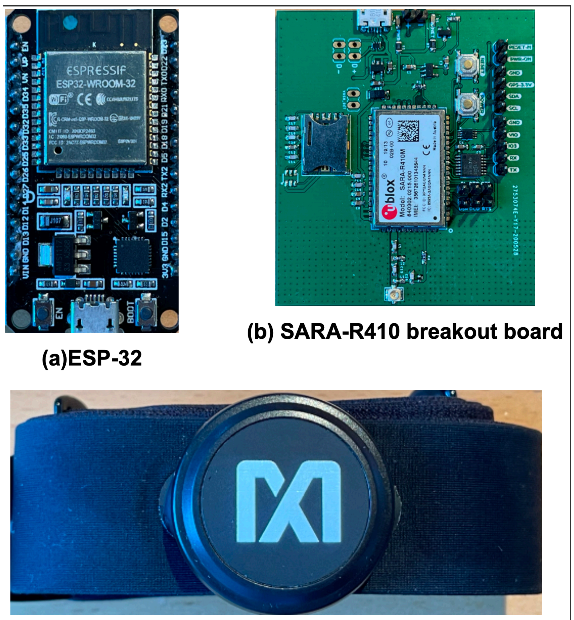
Figure 3.12 shows the output data from the Savvy ECG monitor. The blue graph indicates unfiltered data, whereas the red line indicates filtered data after using a smoothing filter.

3.4.2 Continuous monitoring using Cellular Internet of Things (C-IoT)

Monitoring a patient's heart and showing the data in the form of a continuous graph can bring some insights to the patient viewing the results. However, the real benefits of this monitoring can be achieved if the data can be transferred to a remote server where a medical professional can see the data and understand the health condition of the patient without a in person physical examination. In this section, we try to achieve sending the ECG and HR received by the Savvy and Movesense sensors to a server for further processing. The idea behind this prototype is to connect the wireless ECG and HR monitoring sensor to the ESP-32 and, using LTE-M communication technologies, have this incoming heart data transferred to a remote location for storage and data visualization.

Design of continuous monitoring system

According to the identified requirement in section 3.3.1, and the feature requirement LTE-M was chosen as the preferred primary communication technology in section2.2.3. Figure



(a)ESP-32

(b) SARA-R410 breakout board

(c) Maxim Heart Monitor

Figure 3.13: First prototype components [57]

3.13 shows the components used to design the first prototype.

Figure 3.13 (a) shows the ESP-32 that was used during building the first prototype. 3.13 (c) shows the Movesense ECG and HR monitor.

Figure 3.13 (b) shows the Ublox Saara R410 communication module. A C-IoT communication module connects to the microcontroller using various interfaces (Inter-Integrated Circuit (I2C), Universal Asynchronous Receiver/Transmitter (UART), Serial Peripheral Interface (SPI)). The module is connected to ESP32 using a UART connection in this prototype. SARA-R410 module was chosen particularly because it supports both NB-IoT and LTE-M communication technologies, the small form factor, and firmware that can be controlled using AT commands. The module supports many IoT protocols, including User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Hypertext Transfer Protocol (HTTP), Transport Layer Security (TLS), Constrained Application Protocol (CoAP), and Message Queuing Telemetry Transport (MQTT). This allows for flexibility while sending the user data to the backend systems. The module was fully compatible with the 3GPP rel. 13, that was deployed across Denmark at the time of prototype development.

Figure 3.14 shows the first prototype. The compact design and ability to run the device on Lithium-ion batteries enable the portability of the device. Once powered ON, the devices automatically detects the relevant Movesense ECG and HR sensor and start the setup process. After the setup process is complete, the device starts sending ECG and HR data to the remote server. The device works as a plug-and-play setup, eliminating the need for any external configuration or human intervention during setup as well as during the measurements.

The custom payload format used for monitoring the ECG and HR in realtime involves the utilization of a Movesense sensor. The sensor is configured to monitor and sample heart activity for one second, with a sampling frequency of 128Hz. Each sample recorded by the sensor is a 2-byte signed Hexadecimal (HEX) value. As we are recording at 128Hz, the sensor generates 256 bytes of ECG data. The device processes the received data

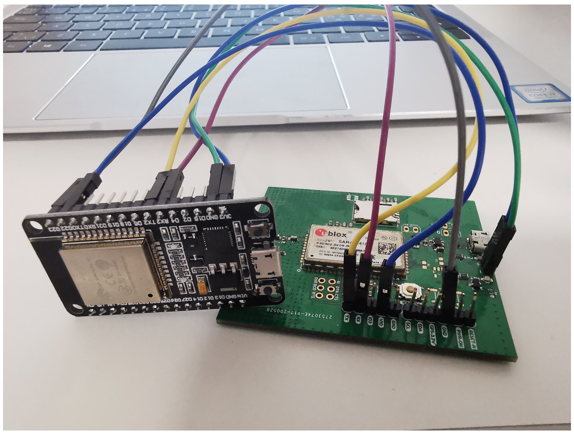


Figure 3.14: First Prototype [58]

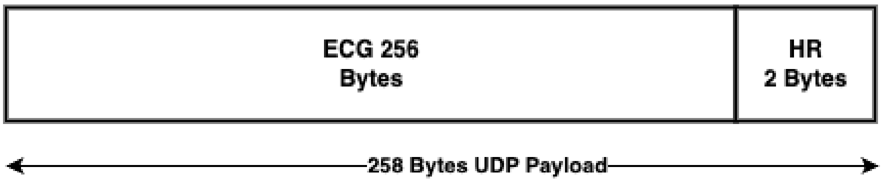


Figure 3.15: Heart Monitor UDP Packet Format (based on[58])

locally and constructs an IP packet, as shown in Figure 3.15. The ECG data occupies 256 bytes, and there are 2 bytes allocated for the average HR value. Once the packet is formed, the device encapsulates it in a UDP header, including all relevant parameters, such as the remote IP address, port number, etc. The device then transmits the UDP packet to the remote server for further processing and storage.

Figure 3.16 shows the complete flowchart of the functioning of the first prototype. When the device is powered on, it starts by initializing all the externally connected peripherals. Once all the modules are powered ON and initialized, the device sends a command to the Ublox SARA-R410 module to connect to the C-IoT network. Since the module connects to multiple C-IoT technologies, the preferred technology choice is coded into the firmware (in this case LTE-M). However, it is also possible to dynamically select the technology based on factors like the best available coverage in the area or bandwidth requirements. After completing the LTE-M network initialization process, the device verifies the connection link and obtains the IP address. If the device encounters any issues with network connection, the ESP-32 resets the module and starts the connection process again. Once the network connection is successful, the device starts the process of connecting to the Movesense heart monitor. In this case, the device is preconfigured with the necessary BLE characteristics, making it easier to initiate the connection. Once the BLE connection is established, the device subscribes to ECG and HR characteristics and that enable BLE notifications. This allows the Movesense ECG and HR sensor to send the data every time new data is available, for this prototype the sampling frequency is one second. This approach is superior to constantly reading values from the sensor every second. It also enables the device to process the ECG and HR and send it to the remote server before the arrival of new data. Once the data is sent to the remote server, the device waits for the new incoming ECG and HR data. This loop continues as long as the BLE connection between the device and the Movesense ECG and HR is active.

Figure 3.17 shows the E2E architecture used for the first prototype. The device prototype

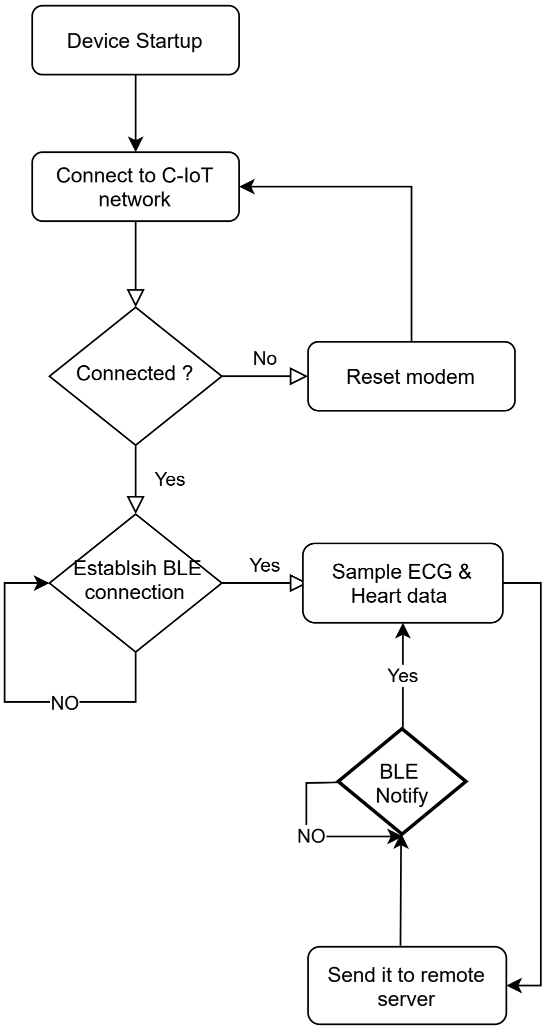


Figure 3.16: Heart Monitor Flowchart

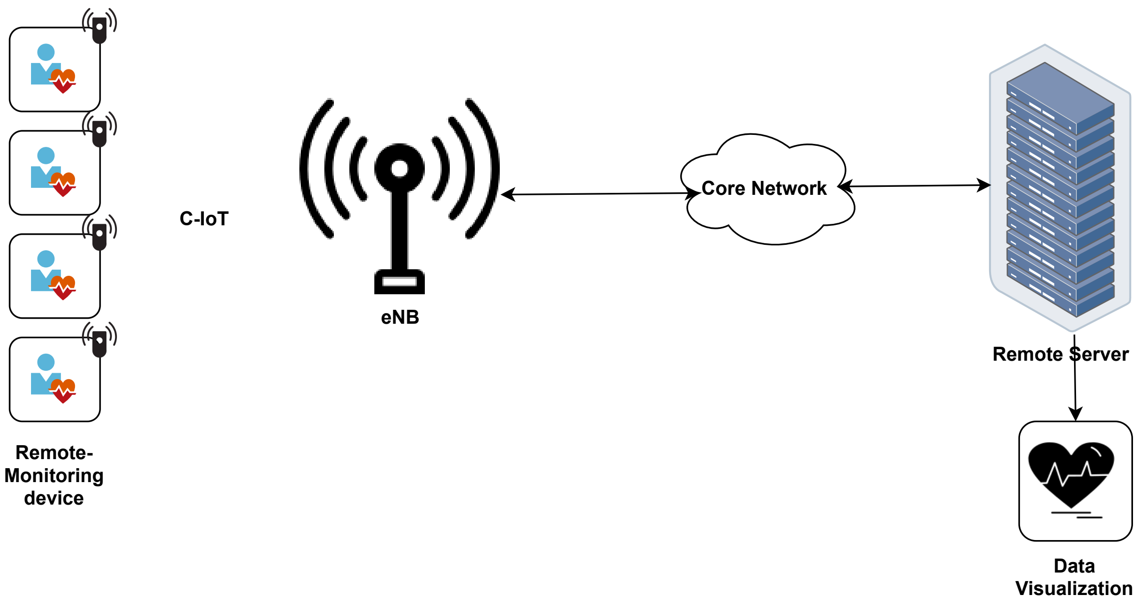


Figure 3.17: Heart Monitor E2E Architecture [57]

in figure 3.14 uses LTE-M to connect to the network and obtain the IP address. Once the device establishes a connection and acquires an IP address, it sends packaged data to a remote server. The data transmission is performed using UDP, a lightweight and connectionless transport protocol. The data is sent to a remote server located in Frankfurt, Germany. The server runs a Python script responsible for handling the payload received from the device. The server processes the data in realtime, extracting relevant information from the payload. The extracted ECG and HR are stored in InfluxDB, a time series database for further consumption[59].

Drawbacks of the first prototype

Although, the design of the first prototype was as per the requirements identified in section 3.3.1 . Several design choices needed further improvements. All the shortcomings of the first prototype are listed below.

1. Although the prototype could run on a battery, the design of the prototype could have been more portable. The prototype's size was way too big for it to be carried around.
2. Another issue with the prototype was that different components were connected using wires. This was not very particle design, given that the ambition in this case was to use the device for continuous monitoring regardless of the location.
3. The prototype was built using off-the-shelf components. It meant there was no control over the design of the device. The components used in the prototype had higher power consumption, e.g., the ESP32 alone was consuming about 240 mA current in active state. The intention was to be able to perform continuous ECG and HR monitoring for a longer time. Higher power consumption meant the patients needed a higher-capacity bulky battery.
4. There was no way to locate the patient in an emergency situation with the current design. Therefore we needed a mechanism to locate the patient's location if required.

Considering all the points above, it was decided to make a new prototype designed.

3.5 Design and Validating second prototype

In this section design and validation of the second prototype is described. The requirements for the second prototype came from the knowledge learned while designing the first prototype and the clinical feedback.

In order to achieve the goal of building this prototype which can receive data from the ECG sensor and send it to a remote backend server in realtime to be visualized following components hardware were needed.

- Microcontroller
- BLE module
- Global Navigation Satellite System (GNSS) Module

- C-IoT communication module
- Printed Circuit Board (PCB) design to mount these components

In this section, all the choices of the components that are used while designing the second prototype are described.

Mirocontroller:

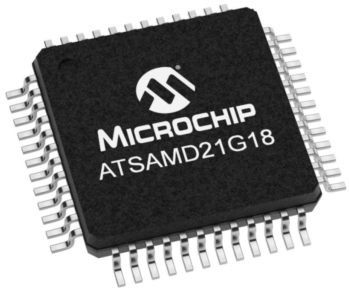


Figure 3.18: Microcontroller ATSAMD21 [60]

A central processing unit that can interface with different peripheral modules such as BLE, C-IoT communication module, and GNSS is needed. Therefore, a microcontroller was needed to handle multiple serial communication channels simultaneously while being power efficient. The initial firmware for prototype one was written in Arduino IDE; therefore, choosing a microcontroller compatible with any of the Arduino bootloaders was a wise option. It was decided to proceed with Microchip ATSAMD21 as the choice for the microcontroller shown in figure 3.18. ATSAMD21 has the following highlighting features which made it a perfect fit for the prototype [60]:

- ATSAMD21 features a small 7mm x 7mm form factor. This allows for a very compact design of the microcontroller.
- ATSAMD21 offers six serial communication interfaces that can be configured to use UART communication protocol and simultaneously handle the communication with all the peripherals, such as BLE, GNSS, and the C-IoT module.
- ATSAMD21 was fully compatible with Arduino MKE Zero bootloader, allowing us to reuse part of the code from the firmware written for the first prototype but at the same time allowing us to write assembly code which can allow us to access the different registers inside the ATSAMD21. This was especially important given the low power requirements of the prototype.
- ATSAMD21 operates between 1.62V and 3.63V, providing flexibility in power supply options and compatibility with various peripheral devices.

BLE module:

In the first prototype, the ESP32 had a built-in BLE protocol stack as well as a transceiver. Unfortunately, this was not the case with the ATSAMD21 microcontroller chosen for the final prototype. Therefore it was necessary to find a suitable external BLE module that could fit the design criteria of the prototype. The primary criteria for the BLE module were to have a compact design, support for serial UART communication, and Bluetooth 5.0 (which was the latest when designing the prototype). Considering all these criteria, Ublox NINA-B312 was chosen as the BLE module shown in figure 3.19 [61].

Figure 3.19: BLE Module Ublox NINA-B312 [61]

- Ublox NINA-B312 supports Bluetooth 5.0 and is backward compatible with previous Bluetooth versions. Its low power consumption allows it to be used in battery-powered IoT applications.
- Ublox NINA-B312 features a compact size of 10 mm x15 mm, allowing including built-in antenna. This allows for the compact form factor of the final PCB.
- Ublox NINA-B312 provides UART and I2C interfaces to integrate host devices and microcontrollers seamlessly.
- Ublox NINA-B312 runs on a 1.7V to 3.6V similar to that of ATSAMD21, making the power design of the PCB simpler.

Global Navigation Satellite System (GNSS) Module:

In the first prototype, no location module was used, but it was later decided that the location of the patient might be very relevant in case of an emergency. Therefore it was necessary to find a GNSS module that could be used for such purposes. The main criteria for the choice were to find a GNSS module that features a compact design at the same time, is more power efficient. Based on the design criteria, it was decided to use a TESEO-LIV3R GNSS module design by ST Microelectronics (STM). Figure 3.20 shows the overall form factor of the GNSS module. Some of the key features of the module are described below [62]:

Figure 3.20: GNSS Module ST Microelectronics TESEO-LIV3R [63]

- STM TESEO-LIV3R GNSS module supports multiple satellite constellations (GPS,

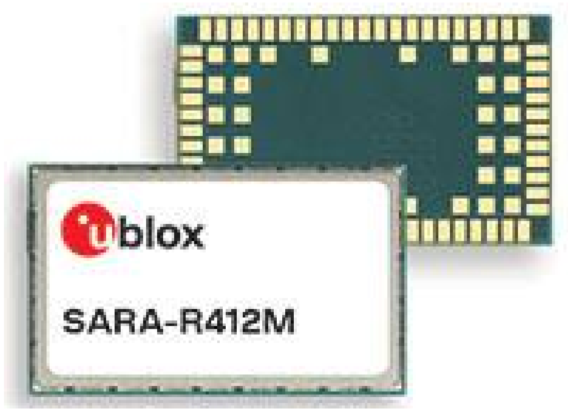


Figure 3.21: C-IoT Module Ublox SARA-R412M [64]

GLONASS, Galileo, and BeiDou) to determine the location. Multi-Constellation support improves the availability and accuracy of the location.

- Regarding receiver sensitivity STM TESEO-LIV3R GNSS module is on the higher end of the spectrum, supporting up to -163dBm. This allows faster location calculation in areas where there is limited visibility of the sky.
- STM TESEO-LIV3R GNSS module features a small form factor of 9.7mm x 10.1mm. This allows for the compact design of the overall PCB.
- STM TESEO-LIV3R GNSS supports various serial communication interfaces, including UART, I2C, and SPI. This allows easy integration with the ATSAMD21 microcontroller chosen for this prototype.
- The operating voltage range for the STM TESEO-LIV3R GNSS module is between 2.1V to 4.3V, making it compatible with the overall PCB power design.

Cellular Internet of Things (C-IoT) communication module:

Ublox SARA-R412M shown in figure 3.21 is from the same family of modules as that of Ublox SARA-R410M. Both of these modules share a similar set of features, except the fact that SARA-R412M also supports 2G fallback. This was added as a feature for this prototype. Therefore, if required 2G was expected to be used as a fallback technology to send data across the mobile network to a remote server. This feture was not used in the actual firmware, instead a BLE fallback feature was developed. Some of the features of Ublox SARA-R412M are described below:

- Ublox SARA-R412M is a multi-region cellular communication module with support for most of the LTE frequencies worldwide.
- Ublox SARA-R412M supports the use of NB-IoT, LTE-M, and 2G (primarily as a fallback technology).
- The form factor of Ublox SARA-R412M is 16mm x 26mm.
- Ublox SARA-R412M does not support the new eSIM standards, forcing the PCB to have an external SIM slot. This increases the overall size of the final PCB.

Low power consumption, support for multiple IoT protocol (UDP, TCP, HTTPS, MQTT, CoAP, etc.), easy-to-use AT command interface, and the possibility of Firmware Over-The-Air (FOTA) updates makes this module ideal to be used in this prototype [64].

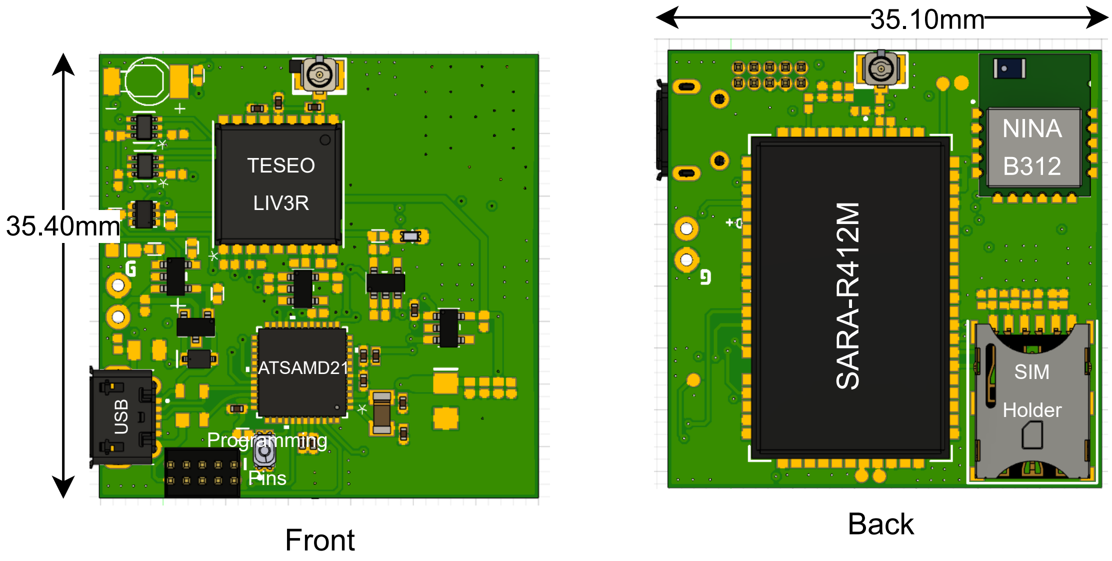


Figure 3.22: PCB Design for second prototype

PCB Design:

Figure 3.22 shows the Front and Back design of the PCB developed during this prototype. All the different components chosen for this prototype's development are highlighted on the drawing. As it can be seen in the images, the design of the PCB is very compact 35.40mm x 35.10mm.

Figure 3.23 shows the simplified overview of the complete working of the second prototype. The functionality can be divided into two parts. Part one is where the device tries to establish a BLE connection with the Movesense ECG and HR monitor, and Part two is where it starts sending the ECG and HR data to the remote server using the LTE-M network.

As soon as the device is powered ON, it starts by initializing all the General Purpose Input/Output (GPIO) pins that will give power to the BLE and C-IoT modules. Once the initial device setup is complete, the device resets the SARA-R412 C-IoT communication module and initiates the LTE-M connection. The device waits until the LTE-M connection is established. Once the LTE-M connection is established, it calls for a function that allows the BLE module on board to connect with the Android app eMed. The app's functionality is described in section 4.3.2. The device waits for 30 seconds and checks for the bonding status. If the BLE app bonding is successful, it continues; otherwise, it resets the BLE device. After that, the BLE mode of the device is changed to central and verified. The central mode in BLE enables the prototype device to scan for the advertisements from the Movesense ECG and HR monitor. Once the device is verified to be in the central mode, the device starts the discovery process to look for the Movesense ECG sensor. Here the discovery process is performed three times for a duration of 1 second each. If the sensor is not discovered, the BLE module is reset. When the Movesense ECG module is discovered, the device initiates the Asynchronous Connection-oriented Logical transport (ACL) connection processes and looks for the BLE services. Once the relevant services are found, the ECG and HR characteristics are written. This tells the Movesens sensors that the device requests two measurements. After that, the relevant characteristics are subscribed for notification.

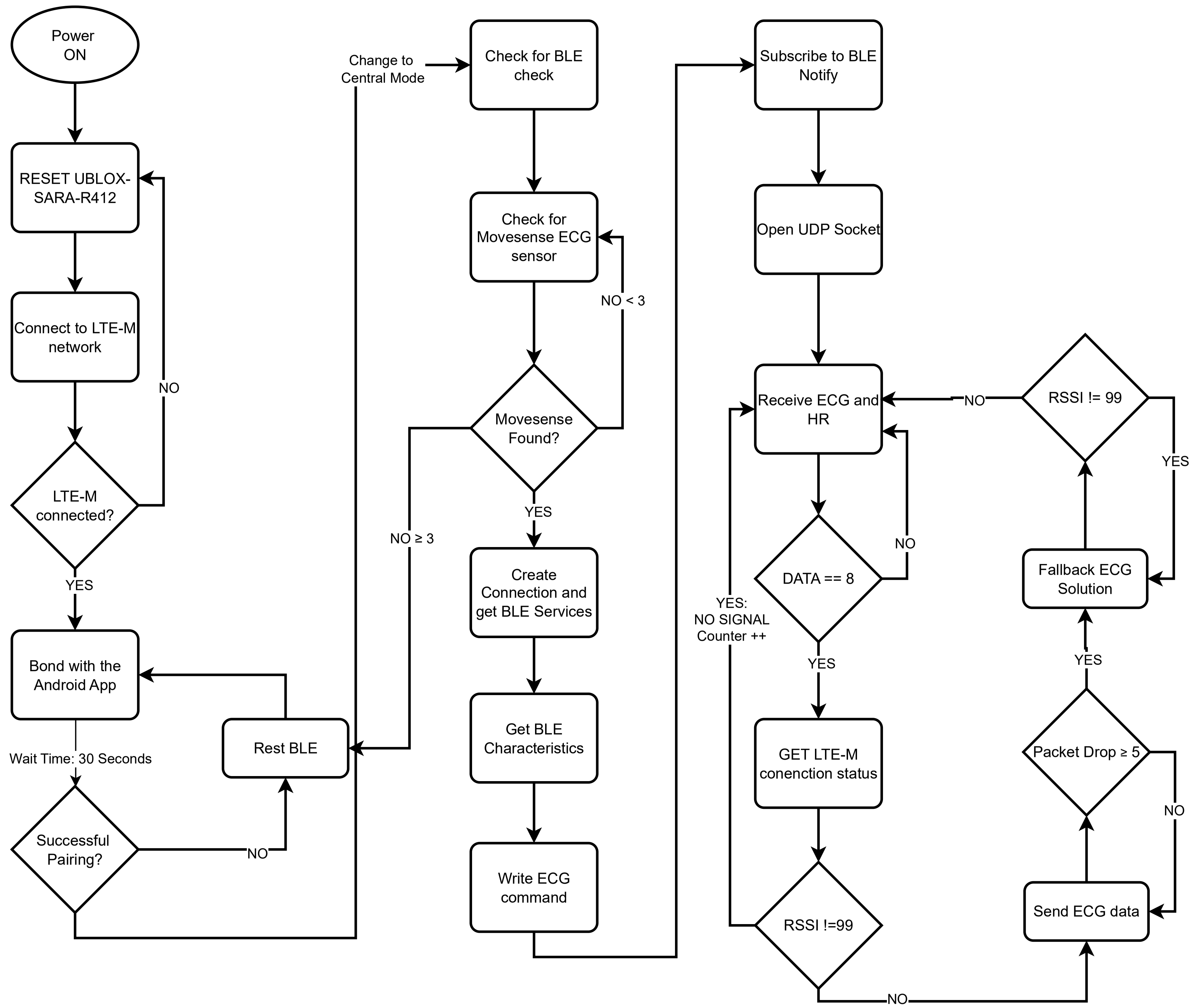


Figure 3.23: Overview of working of Second prototype (based on [58]

Once the BLE setup is complete, the device opens a UDP socket on the SARA-R412 communication module. This UDP socket sends the ECG and HR data to the remote server.

Once the BLE and LTE-M setups are complete, the device starts accepting and processing data from the Movesense ECG and HR monitor. The Movesense ECG and HR monitor is set up to sample heart activity at 128 Hz, and each data packet received from the sensor contains 16 measurement samples. Therefore, in total, the Movesense sensor sends eight packets. The device buffers the incoming packets until the packet count equals eight. Once eight packets are received, the device makes a final check whether the LTE-M network is still connected or not. If the device has LTE-M connectivity, the data is sent to the remote server using the UDP socket. Otherwise, the no signal counter is incremented. Meanwhile, the device has started preparing the next packet; once this packet is ready to be sent, the device will again check for LTE-M connectivity and send the packet over. The fallback solution takes over if five consecutive packets are dropped due to no signal. In the fallback solution, the device will notify the eMed app to connect with the Movesense ECG and HR monitor monitor and continue recording the ECG data in full resolution.

While the system is in the fallback solution, the IoT prototype device will try to regain the LTE-M connection, and if it is successful, it will restart the complete BLE pairing process. After the BLE process is complete, the device will again start sending the data using LTE-M.

3.6 Feedback from the clinical side

As a part of the Phase II and Phase III of the PD, the end device with the ECG and HR sensor was shown to different groups of individuals, including medical doctors, medical professionals, and researchers in healthcare & telerehabilitation. The feedback received by different groups is motioned below:

- The size and the design of the end device PCB are very positive. At approximately 3.5mm x 3.5 mm, it is small enough to be used in various remote monitoring scenarios. The BLE link with the ECG sensors is another positive sign in the design of the end device.
- It was recommended to design a plastic enclosure in different shapes to house the PCB design. The most common suggestions were to design the enclosure in the form of a wristwatch, and the other was to design it as a waist belt clip.
- Another feedback related to the end device enclosure was to design the housing, which is IP68 certified. This is meant to make the device dust and water-resistant.
- To add a display to show the ECG and HR measurements on the device, and when the device is not measuring the ECG or HR, then use the screen to show the date and time. This feature will help keep the device more discrete and more acceptable in society.
- The current PCB design used a micro-USB charging port; it was recommended to change it to more commonly found charging ports like USB-type C or to use include Qi wireless charging.

4 Server and Backend

4.1 Data visualisation background

In several studies around cardiac rehabilitation, physical activity has been found to have a significant impact in reducing mortality rates in Cardiovascular Disease (CVD) [10]. Therefore it was decided to include a step counter to measure the daily activity; based on the Future Patient Telerehabilitation (FPT) project's research output, it was decided to collect the blood pressure and sleep data from the individual patients [18]. Alongside these once-a-day measurement data points, a particular feature was developed to receive the continuous Electrocardiogram (ECG) and Heart Rate (HR) data. In addition to these features, there was a need to develop the backend system, which is in line with the system requirements highlighted in Chapter 1. This included developing a data storage system to make data available for later consumption, visualize continuous ECG and HR, visualize once a day measurements, interfacing with other backend systems, different ways to export data, data The system needed to support methods to request the data so that it could be used for further processing. A visualization feature will allow CVD patients and their loved ones to consume the recorded data and monitor the progress of the individuals. The last requirement was using Machine Learning (ML) to assist medical professionals in detecting abnormalities in the recorded data.

The rest of the chapter is divided into three sections: Data collection and Storage, Data visualization, and ML Algorithms, describing the functionality developed during this project to meet the requirements set for this part of the system as an outcome of the different phases of the Participatory Design methodology.

4.2 Data collection and storage

Similar to how the end device designed during this PhD went through multiple iterations and Participatory Design, the data collection and storage system was designed using the same principle. The system underwent three revisions, which are discussed in this section. The primary aim behind developing such a system was to be able to store the incoming ECG and HR data from the end device.

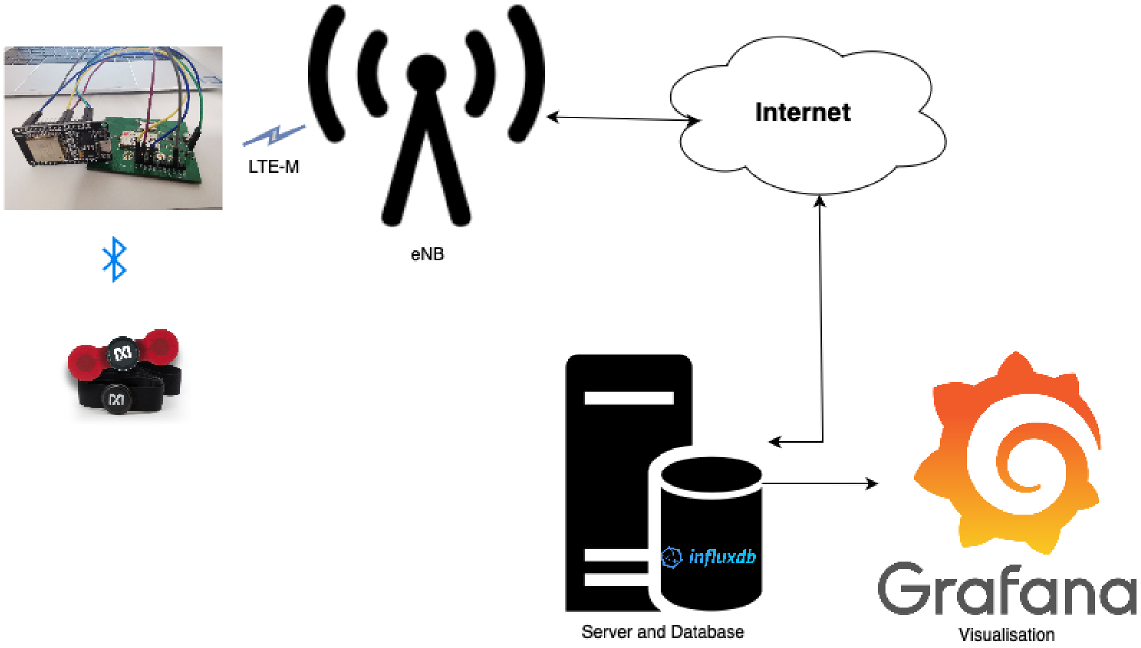


Figure 4.1: system Overview Prototype1

4.2.1 Data Collection and Storage System: prototype 1

In this section, we will go over the initial design of the End-to-End (E2E) architecture of the prototype data collection and storage system. From the beginning of the project, the ambition has been to send the collected heart data to a remote server for analysis by medical professionals. Step one in achieving this goal was to create a digital infrastructure that can accept the incoming data from the end device, store it, process it, and visualize it.

Figure 4.1 shows the E2E architecture used to develop the first data collection and visualization system prototype. As can be seen from the figure, the data from the Movesense ECG and HR monitor was sent to the end device. The data was packaged in the device and encapsulated in the UDP packet, and sent to the server using LTE-M. The after the data enters the Mobile Network Operators (MNO) network, it is forwarded to the online server hosted in Frankfurt, Germany. A script runs at the server that accepts the incoming data and stores it in the InfluxDB database. ECG and HR, are measured in reference to the time. Therefore it was essential to find a database that can handle time series data and make it available for realtime consumption. InfulxDB is an open-source time series database. InfluxDB can help in time stamping the data as soon as it is being inserted into the database and make the data available for realtime consumption [59]. InfluxDB is widely used in the IoT environment to store sensor data, performance monitoring data, etc. InfluxDB also provides different techniques to handle high volumes of data by allowing better data compression, faster queries, and scalability [65].

Once the data was inserted into the database, it was fetched by Grafana. Grafana is also an open-source data visualization and monitoring platform. Grafana supports built-in integration with the InfluxDB database. This allowed for a much faster integration between the data storage and visualization [66].

Figure 4.2 shows the ECG graph being printed in realtime. As can be noticed from the graph, the different peaks of the plot are not synchronized. This was because Grafana

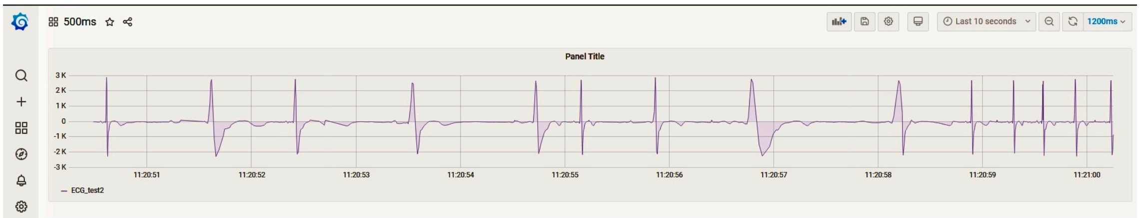


Figure 4.2: ECG visualisation using Grafana

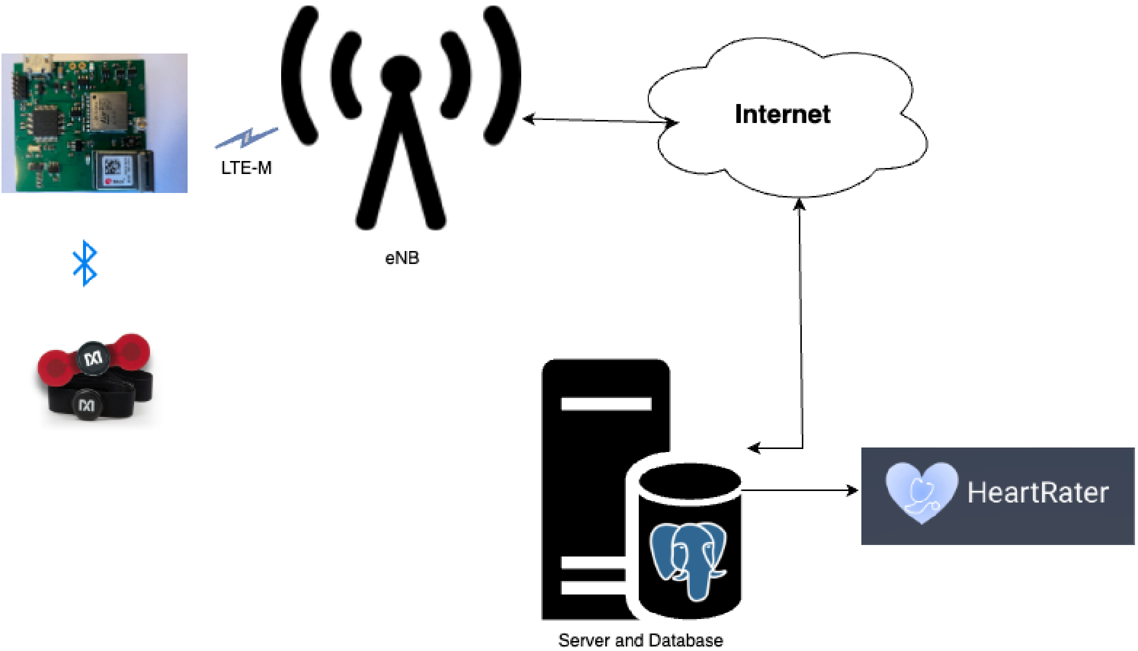


Figure 4.3: System overview prototype 2

prints the data according to the timestamp found in the InfluxDB database. This timestamp is inserted automatically by the database when the decoded ECG and HR data is inserted into the table. The heart data was sent to the server using the data structure described in Figure 3.15. The data measurement was performed by sampling the heart activity 128 times for 1 second and sent to the server. There was always an error introduced in storing data with just one timestamp. To an extent, this was solved by artificially adding delays while saving every measurement to the database. Another timing error was introduced because the server's data arrival rate was not constant. This meant that no two packets were arriving at the same time. This can be seen from the ECG plot that the distance between two consecutive R peaks is not constant.

4.2.2 Data Collection and Storage System: prototype 2

This section goes over the second version of the data collection and storage system.

Figure 4.3 shows the overview of prototype 2. There are the following changes made to the system:

- The system uses a new and improved end device prototype with the Printed Circuit Board (PCB) designed at DTU.

- The system uses PostgreSQL as the database instead of using InfluxDB [67]. This was mainly to accommodate the changes in the backend architecture.
- Development of entirely new front-end data visualization webUI hosted on heartrater.live domain. The server was hosted on a remote server in Frankfurt, Germany.

The data flow in the case of the system overview described in Figure 4.3 is very similar to that of in the case of 4.1. The data from the Movesense ECG and HR monitor is sent to the new end device prototype, where the data is encapsulated into a UDP packet and sent to the server using LTE-M. Once the data reaches the server, it is stored in the PostgreSQL database. PostgreSQL is a relational database management system that is robust, compatible with SQL standards, and scalable [67]. The move from InfluxDB to PostgreSQL was primarily done to make the system more generic. In the case of influxDB, the data structure was more rigid and used only one object containing all the measurement types. Since the project was in the research stage, adding and removing data types from the table format would have made it more difficult and time-consuming. In other words, it would have made it difficult to add/remove columns from the tables stored in the database as the data structure kept on evolving throughout the duration of this project.

Figure 4.4 shows the software architecture of the deployed solution. As can be seen from the figure, the architecture is divided into three main categories Database, Backend, and Frontend. The primary reason for dividing the solution into multiple parts was to make it as generic as possible. This division allows for easy extension of the existing deployed solution and swapping different components to a different technology.

- **Database:** As it can be seen from the figure 4.4 the database used in this prototype is PostgrSQL. The database consists of three tables patient_data_1, ecg, and thresholds. Figure 4.5 shows the new data structure used in tables patient_data_1. The daily measurements from the patients are stored in tables patient_data_1. The realtime heart data sent by the prototype end device is stored in the ecg table. The data stored in the threshold table sets the threshold values for different measurement types stored in table patient_data_1. As the name suggests, the values from this table are stored to highlight the different thresholds by the medical professionals for each patient ID.
- **Backend:** The system's backend is divided into three components: Repositories, Models, and Controllers.
  - Repositories: Java API repositories are used for queering the data from the PostgreSQL database. Repositories allow controllers to access data from the database in the format defined in the data models.
  - Model: The model layer consists of the definition of objects and their structure that the controller uses to map the data and repositories to query the data. The model's definition must match the table format containing the data.
  - Controllers: The primary purpose of the controller is to expose the APIs which can be used by different web applications to query the data. These APIs are used by the front end of this web application.
- **Forntend:** The primary objective of the frontend is to visualize the data exposed by the different APIs in the backend. The frontend has two data services measurement data service and frontend data service. Both of these data services send Hypertext Transfer Protocol (HTTP) requests using Axios and node.js [69, 70]. Data

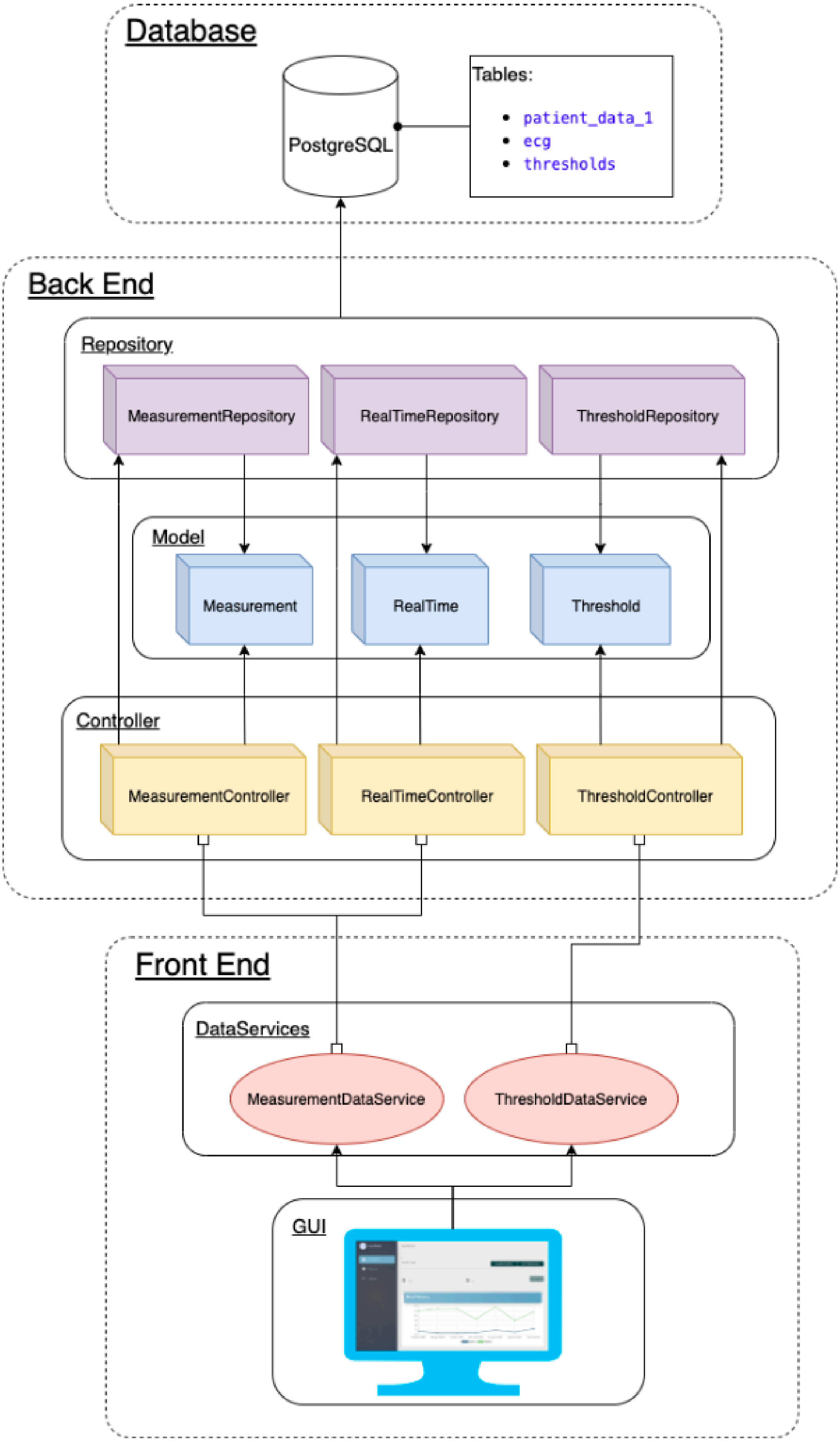
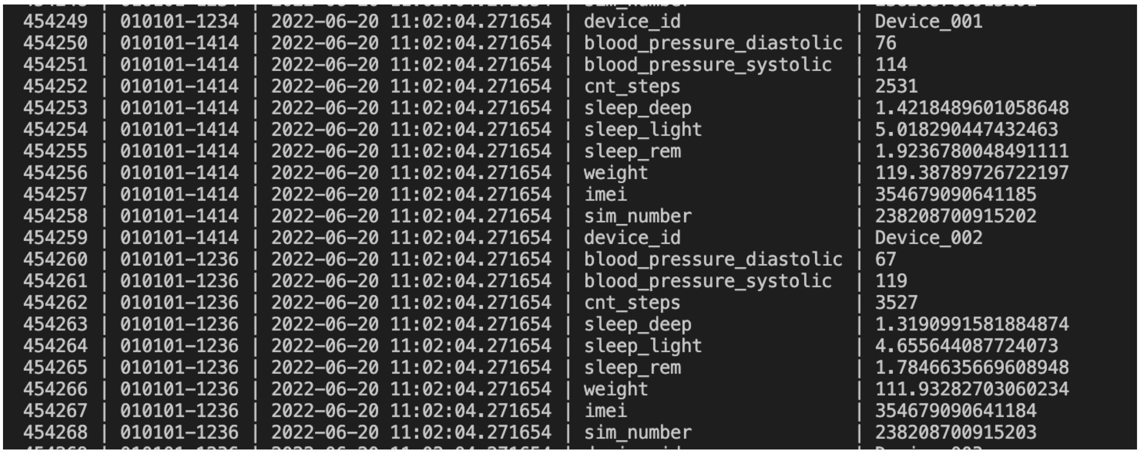


Figure 4.4: System Architecture V1 [68]



454249	010101-1234	2022-06-20 11:02:04.271654	device_id	Device_001
454250	010101-1414	2022-06-20 11:02:04.271654	blood_pressure_diastolic	76
454251	010101-1414	2022-06-20 11:02:04.271654	blood_pressure_systolic	114
454252	010101-1414	2022-06-20 11:02:04.271654	cnt_steps	2531
454253	010101-1414	2022-06-20 11:02:04.271654	sleep_deep	1.4218489601058648
454254	010101-1414	2022-06-20 11:02:04.271654	sleep_light	5.018290447432463
454255	010101-1414	2022-06-20 11:02:04.271654	sleep_rem	1.9236780048491111
454256	010101-1414	2022-06-20 11:02:04.271654	weight	119.38789726722197
454257	010101-1414	2022-06-20 11:02:04.271654	imei	354679090641185
454258	010101-1414	2022-06-20 11:02:04.271654	sim_number	238208700915202
454259	010101-1414	2022-06-20 11:02:04.271654	device_id	Device_002
454260	010101-1236	2022-06-20 11:02:04.271654	blood_pressure_diastolic	67
454261	010101-1236	2022-06-20 11:02:04.271654	blood_pressure_systolic	119
454262	010101-1236	2022-06-20 11:02:04.271654	cnt_steps	3527
454263	010101-1236	2022-06-20 11:02:04.271654	sleep_deep	1.3190991581884874
454264	010101-1236	2022-06-20 11:02:04.271654	sleep_light	4.655644087724073
454265	010101-1236	2022-06-20 11:02:04.271654	sleep_rem	1.7846635669608948
454266	010101-1236	2022-06-20 11:02:04.271654	weight	111.93282703060234
454267	010101-1236	2022-06-20 11:02:04.271654	imei	354679090641184
454268	010101-1236	2022-06-20 11:02:04.271654	sim_number	238208700915203

Figure 4.5: Measurements with Mocked with new data structure (based on [68]

requested by these data services is then displayed in the Graphical User Interface (GUI) described in section 4.3

4.2.3 Data Collection and Storage System: prototype 3

The third prototype of the Data Collection and Storage system further added new functionality for collecting the data from other sensors used in the project, such as pedometers and sleep sensors. This implementation was done in collaboration with Aalborg University, where the data from Fitbit and Emfit was stored. This data was fetched by the heartrater.live system periodically using REST APIs provided by Aalborg University. Following are the new capabilities added in this version of the system prototype:

- The entire heartrater system was moved from the data center in Frankfurt, Germany, to a server infrastructure created for this project at DTU. The server where the system is hosted is shown in figure 4.6.
- Adding the ability to get once-a-day data like, step count, sleep data, etc., from the Aalborg University database.
- Adding the ability for individual patients to view the data generated by them in a mobile app. This feature was added based on the feedback received from the medical professionals for this project.
- Ability to classify the incoming ECG in realtime to detect abnormalities in the measurements and raise a flag on the heartrater.live web portal.

Figure 4.7 shows further improvements to the system's software architecture. Following are the new features added since System Architecture V1.

- In the database, a few more tables were added, mainly ecg_float and ml_output. ecg_float table, as the name suggests, was added to store the floating value of the incoming ECG data. This was done due to inconsistent data plotting of the continuous ECG data from the prototype end device. Similarly, the ml_output table was used for storing the output from the classification model developed to detect the abnormal ECG data.
- This system architecture also added the capability of fetching the real measurement

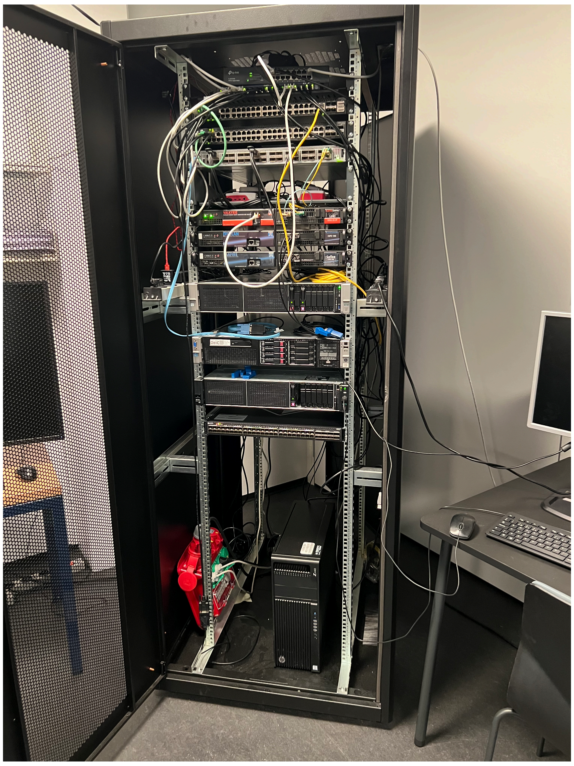


Figure 4.6: DTU Server Hosting Environment

values from an external server hosted at Aalborg University. This mainly included fetching real data generated using different sensors such as Fitbit, Emfit sleep sensor, blood pressure, etc.

- Addition of new functionality also needed modification and improvements to the Backend. New repositories, data models, and controllers were added to handle new data points.
- Improvements to the GUI of the system. e.g., improvements to the headers, the ability to delete thresholds, etc.

Figure 4.8 shows the overview of the entire system. The main feature developed during this system version was the ability to classify the incoming ECG data using a ML model developed on the incoming ECG data from a healthy person. Development of this classification ML model is described in brief in section 4.4. The incoming data was stored in the PostgreSQL database and fed to the ML model simultaneously. The ML model was designed to classify the ECG data and to give a binary output of 0 or 1. This output from the ML model was then stored in the ml_output database; from there, it was displayed on the system's Frontend.

Another functionality added in this prototype version was the ability to consume the data stored on the server and database through an Android App. The design of the Android app is described in section 4.3. The app was mainly designed to be used by individual patients. The app and the prototype end device implemented a unique fallback feature, allowing us to establish a Fallback Bluetooth Low Energy (BLE) link. The purpose of this link was to create a functionality to store data on the Android app when the primary method of data communication was unavailable. Once the prototype end device detected no signal on the LTE-M network, it could send a message to the Android app saying it

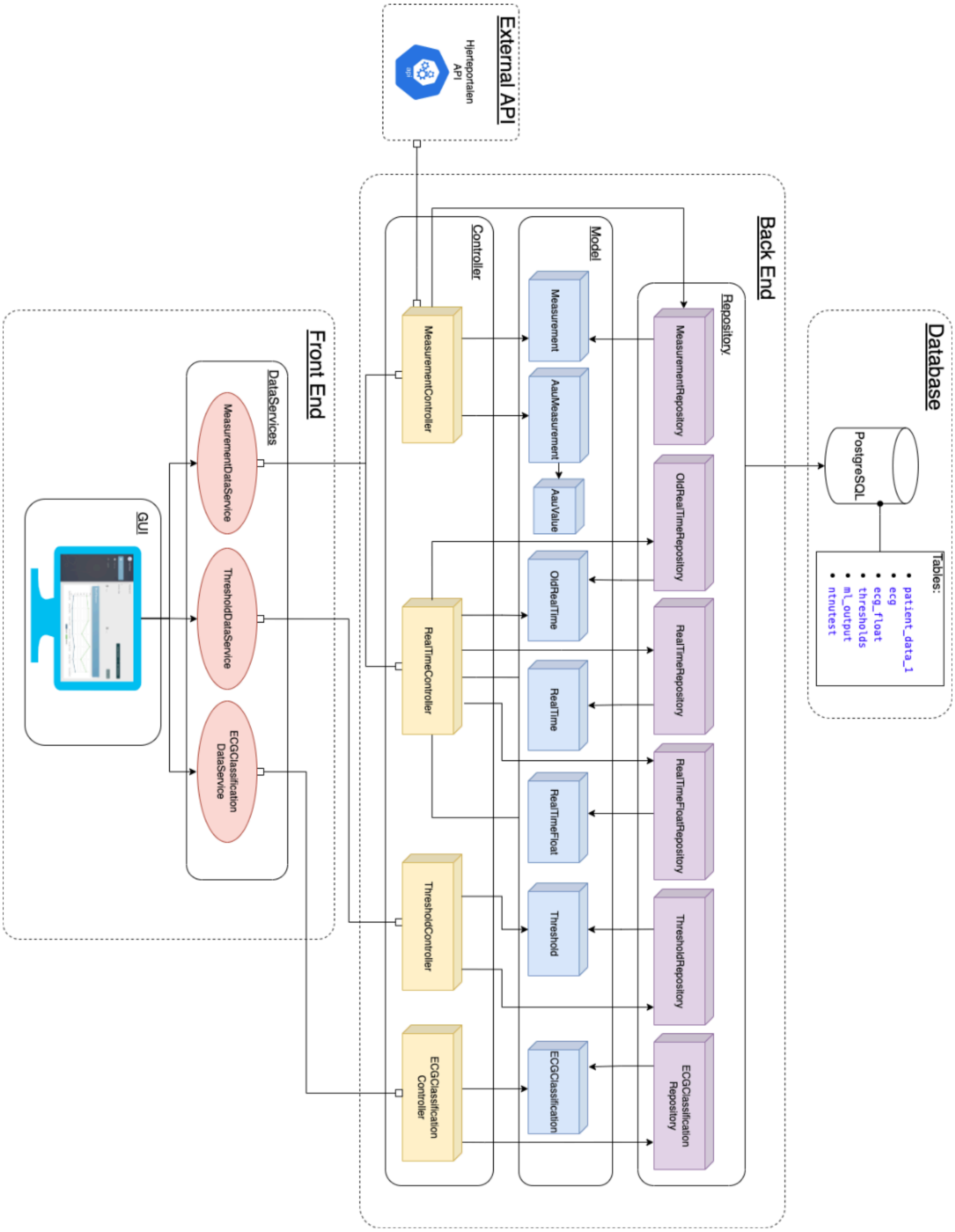


Figure 4.7: System Architecture V2 (based on [68])

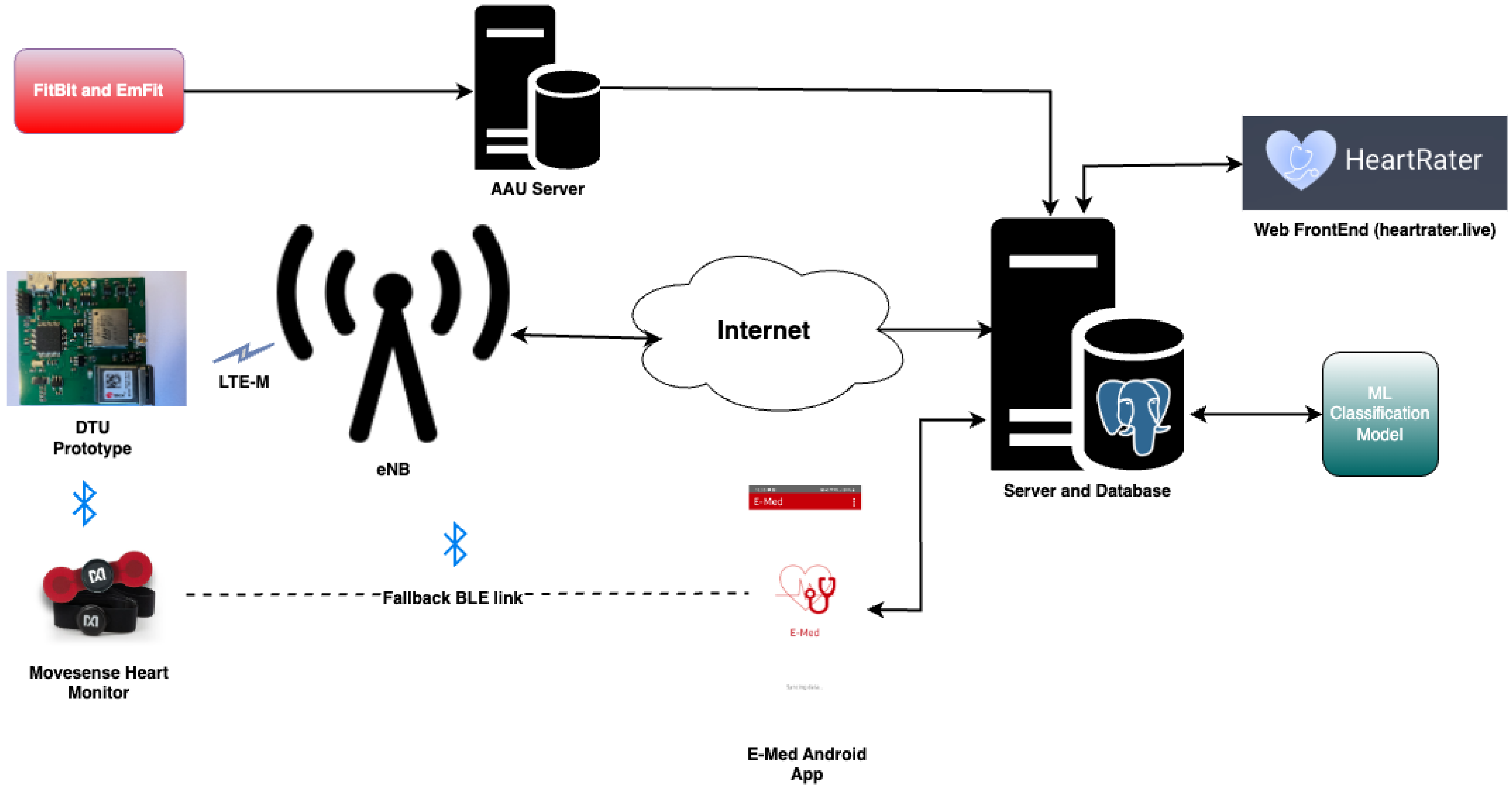


Figure 4.8: System Overview Prototype 3

can directly connect with the Movesence ECG and HR monitor to avoid data loss. An additional API was created, which was designed to accept the data directly from a 3rd party application to add the data to the system.

4.3 Data visualisation

This section of the thesis describes the different visualization interfaces deployed during this project. The overall idea behind the project was to develop a continuous ECG and HR monitoring system that can be used for continuous monitoring patients with Cardiovascular Disease (CVD) from a distance and to view the results using web portals. The idea behind designing a web portal is to provide a holistic picture of a patient's health condition to a medical professional. This meant that a portal was designed to be used for accessing the data from once-a-day measurements, realtime ECG data, and HR. While discussing with CVD patients and the medical professionals, it was discovered that it could be beneficial to develop two visualization systems. One is where the medical professionals can view data from all the patients using the web portal system, and the second is where the patients being monitored can look at their own data. It was then decided to build an Android application to fetch the data stored on the heartrater backend using the developed REST APIs. The decision to develop an Android app was taken later in the project's development based on the medical professional's feedback, especially around improving the Quality of Life (QoL) of the CVD patients.

Therefore, the project focused on developing two visualization tools a web portal for medical professionals to monitor multiple patients using the system and an Android app for individual patients to monitor their own progress. Both tools were developed in collaboration with student projects at the Technical University of Denmark (DTU).

1. Development of API and User Interface (UI) for Visualization of Heart Sensor Data [68].

2. Remote monitoring of heart patients using 5G & IoT [58]

The web portal (heartrater.live) and the android app (e-Med) were validated by conducting tests with several fellow students, performing social surveys, receiving feedback from summer schools, and feedback from medical doctors, medical technicians, and professors in the healthcare domain.

The section is divided into two parts, an overview of the web portal (heartrater.live) and an overview of the Android app (e-Med). The development of both visualization tools is described in detail in the student projects

4.3.1 Web Portal: Heartrater.live

Based on the outcome of Packet Drop, it was decided to have the measurements divided into two categories. The first category was the once-a-day measurements such as step count, sleep, blood pressure, etc., and the second category was the continuous HR & ECG. The once-a-day measurements were finalized at the end of the project. Therefore the project kept using the existing data points identified from the Future Patient Telerehabilitation (FPT) project [6].

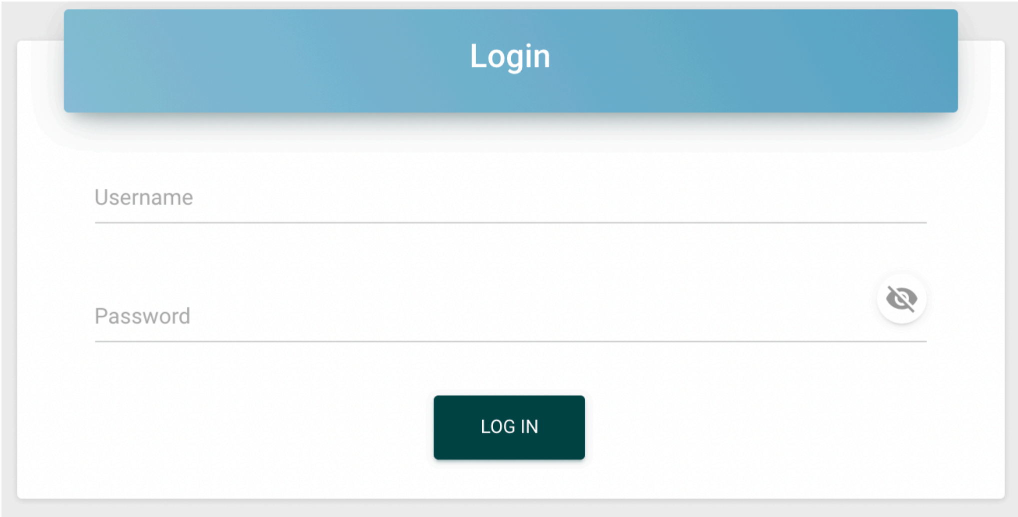


Figure 4.9: Heartrater.live Login page [68]

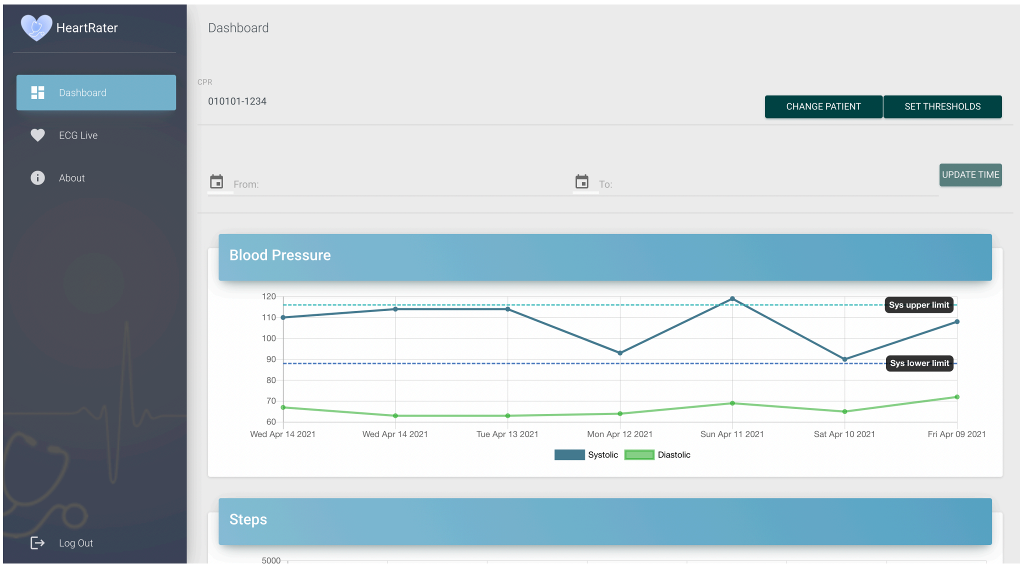


Figure 4.10: Heartrater.live daily measurements page [68]

Figure 4.9 shows the login page for the web portal haertrater.live. Similarly, Figure 4.10 shows the page with daily measurements. In this version of heartrater.live web portal, the daily measurements included blood pressure, steps, and sleep values.

Figure 4.11 shows the page where the medical professionals can set the threshold values for each daily measurement. These values are dynamic and unique per patient. These values were also deletable in the last prototype of the web portal.

4.12 shows the ECG data monitoring page. This page shows the realtime ECG, HR, and ML output. On this page, there is also an added functionality of switching between a patient's live and recorded ECG. The thought behind having this toggle button was to build a functionality where an ML model can already carve out an irregular patterns of ECG for medical professionals to be examined later. Another feature which was added at the bottom of the page was the ability to pause the ECG graph, this feature was added after receiving feedback from the medical professionals about how they read the ECG patterns.

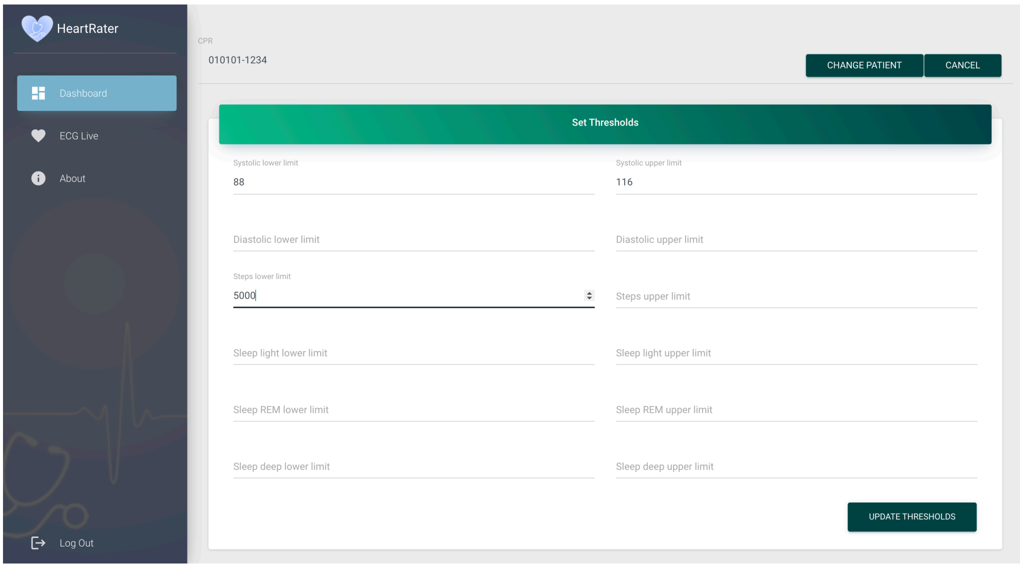


Figure 4.11: Hearrater.live Threshold Page [68]

Figure 4.12: Heartrater.live live monitoring page [68]

4.3.2 Android app e-Med

The Android app e-Med was developed for individual patients to view their health data, including once-a-day measurements and the realtime ECG and Heartrate data. The app was developed using the existing REST APIs developed in section 4.2. The app added the functionality to connect directly to the Movesense ECG and HR monitor using BLE. The primary reason for developing such functionality was to ensure the app can continue monitoring the heart if the cellular connection using LTE-M drops out.

Figure 4.13 shows the different pages designed during the development of the Android app. Figure 4.13 (a) is presented when the app is starting up for the first. In this case, the app is performing a sync with the heartrater.live backend and fetching new and updated information about the daily measurement of the patient. Figure 4.13 (b) presents the choice between doing local monitoring using the Movesense ECG and HR monitor or monitoring through the REST APIs. When the REST APIs are used, graph data shown in Figure 4.13 (c) is delayed a little. This was done to buffer a few packets, which allows for a smoother viewing experience and helps compensate for the network delays caused by LTE-M. Final figure 4.13 (d) shows the average data of once-a-day measurement values.

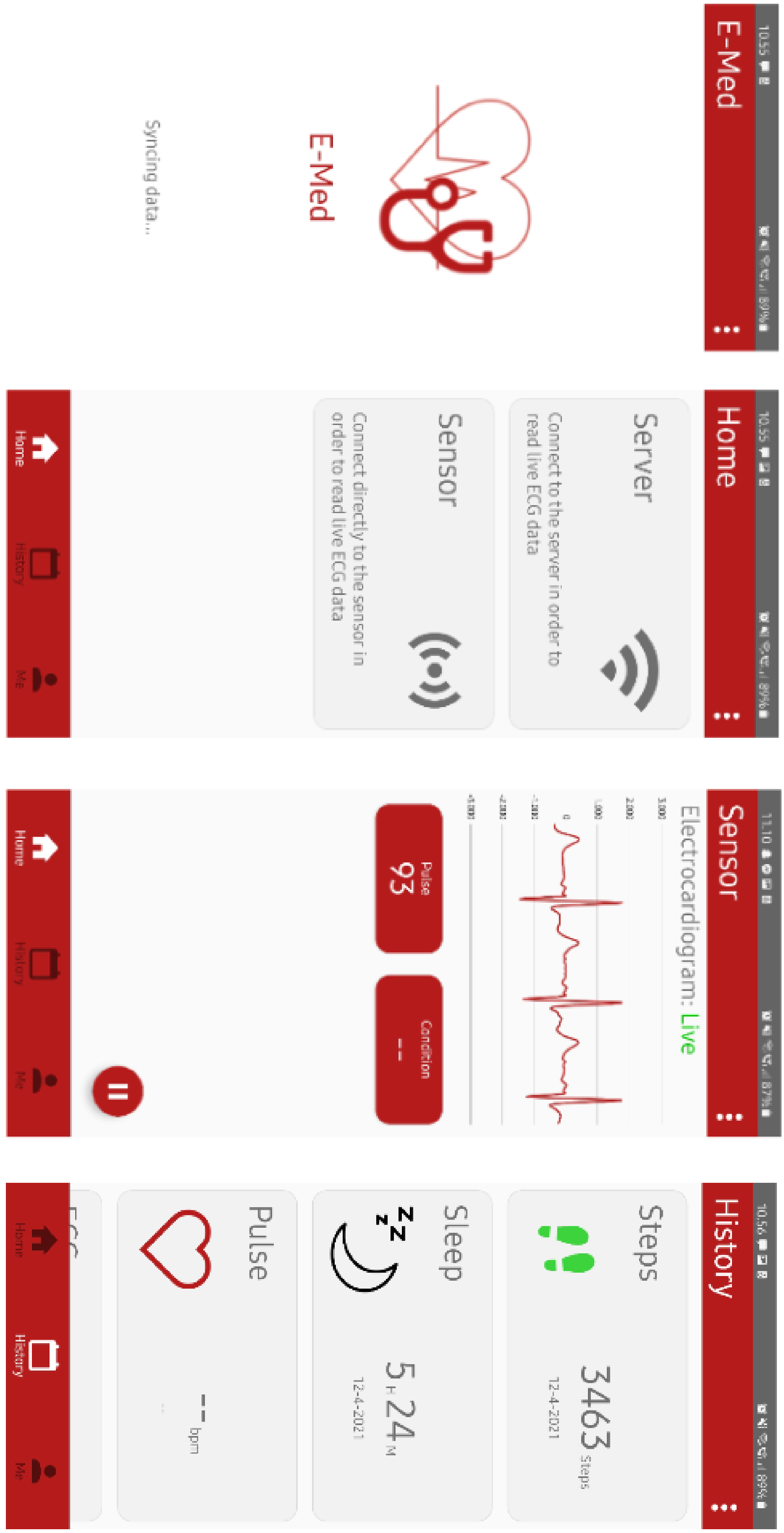
4.4 Machine Learning (ML) Algorithms

One of the critical aspects of developing a continuous ECG and HR monitoring system for CVD patients was to develop an instantaneous feedback system. There was no use in using the system to record the activity of the patients in realtime if there was no realtime monitoring being conducted. Therefore, from the beginning of the project, the ambition was to record the patient activity and monitor it with the help of Machine Learning (ML) to detect abnormalities in the incoming ECG data.

In the initial discussions with the medical professionals, several tasks were identified where using a trained ML model made sense. The identified activities were as follows:

- Predict the once-a-day measurements from the patients. i.e., steps, sleep, weight, etc., to apply to generate early warnings of worsening health conditions based on the observed trends.
- Predict the patient’s heart activity a few seconds in advance to detect early signs of HF.
- To classify incoming once-a-day measurements for signals of health deterioration.
- To classify the incoming ECG signal into normal and abnormal heart activity.

An ML model’s performance depends on the data quality available for training the model. Having access to good quality data may lead to improved accuracy, generalizability, reliability, etc., of the trained model. During this PhD project, accessing real once-a-day measurement data generated by actual patients was difficult due to privacy and General Data Protection Regulation (GDPR) restrictions in handling the health data. Although the once-a-day measurements were begging recorded during the project, the data only contained measurements from one person. At the time of developing the ML model, the measurements were only conducted for 120 days. Therefore, it was decided to continue only with the ECG signal classification model for the time being, where there was a possibility of generating sufficient data, and also try to use the open-source datasets available on the internet.



(a) Splash Page

(b) Home Page

(c) Live ECG

(d) History Page

Figure 4.13: Overview of eMed Android Application [58]

In this project we are dealing with a time series data, i.e. the ECG is recorded continuously over time and the data is in sequence. Therefore, there are several ML that can be used to classify the time series data. Some algorithms that are suited for time series data classification are:

- Long Short-Term Memory (LSTM) Networks
- Convolutional Neural Networks (CNN)
- k-Nearest Neighbors (KNN)
- Random Forest (RF) etc.

Mohamed *et al.* [71] investigated use of Support Vector Machine (SVM), KNN, CNN, and RF in designing automatic classification system for improved detection of minority arrhythmical classes. Ali *et al.* [72] evaluates the performance of different ML such LSTM, CNN, Deep Neural Network (DNN), etc. Yang *et al.* [73] have demonstrated benefits of cross domain transfer of EEG to EEG or EEG to ECG to improve the classification performance of CNN at the same itme reducing the training time required to fully train the model. Malak *et al.* [74] uses KNN and RF to detect abnormalities in the ECG signal, the study resulted in showing RF is slightly better than that of KNN. Elena *et al.* [75] designs a patient specific CNN model to classify ECG from a single led sensor for long term monitoring of patients. Deena *et al.* [76] highlights that accuracy of CNN is much higher in comparison with KNN, RF, and logical regression in classifying time series data from colorimetric sensor. They also discovered that RF and KNN have performed good in classifying sensor data whereas CNN and LSTM have shown much better performance in classifying time series data. Therefore based on these findings, we decided to compare performance of CNN, KNN, and RF while classifying the continuous ECG send by the end device using Movesense ECG and HR monitor. The following section briefly explains the three algorithms used for developing the final ML model.

- **Convolutional Neural Networks (CNN):** CNNs are a feed-forward artificial neural network with primary applications in analyzing visual data. CNNs are very successful in tasks involving image classification, image recognition, and object detection. CNN can be trained by supervised as well as unsupervised training methods. The convolution layer is one of the key components used in constructing the CNN. This layer is mainly used for detecting patterns in an image, such as edges, shapes, etc. CNN also consists of Max Pooling Layers, the primary function of this layer is to reduce the spatial dimensions of the features while preserving the important features of the data. The reduced dimension also reduces the parameter needed to be learned by the CNN. Pooling layers also contribute in making the CNN more robust to the changes in the feature locations in the input images [77, 78].
- **k-Nearest Neighbors (KNN)r:** KNN or the K-nearest neighbor is another commonly used algorithm in classification scenarios. In KNN, in any given training data set, the algorithm assigns the value in the sample data set to the class of values defined by the class of K-nearest neighbors. In other words, KNN looks at the K nearest neighbors to determine (classify) the sample data value. The choice of the K value is crucial in KNN. If the K value is too low, the model can be susceptible to the local variations of the training data. On the other hand, if the value of K is too low, then the model performance may be smoother but can overlook the fine details and patterns in the training data set. The KNN model can be described in two phases. First is the training phase, where the algorithm will try to label the features of the training data set. The second phase is where the prediction takes place, where the

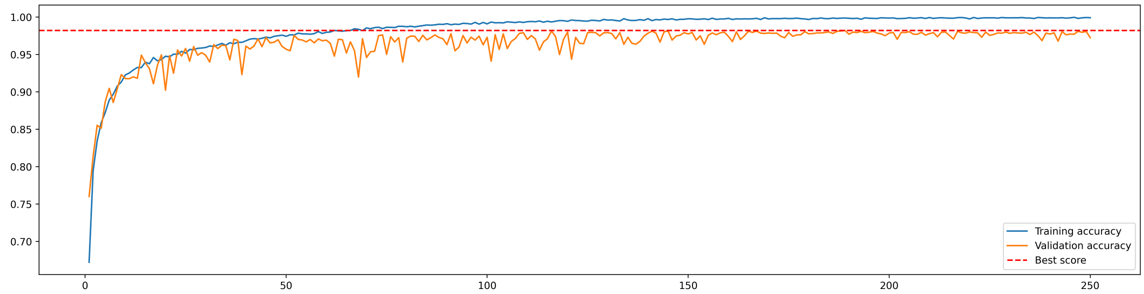


Figure 4.14: CNN classification performance of MIT-BIH arrhythmia aata Set [78]

Euclidean distance between the sample data is calculated, the K nearest neighbors are selected, and sample data is labeled according to the majority class. KNN shines in non-linear complex data distribution but suffers in high dimensional data sets [77, 78].

- **Random Forest (RF):** RF is another popular algorithm used in ML used for data classification and regression. The RF algorithm is based on a decision tree structure where each node corresponds to features, each branch corresponds to a possible outcome, and each leaf corresponds to a classified label and regression value. The RF algorithm is constructed of multiple unrelated and random decision trees. The RF algorithm can be divided into three main phases. The first phase is the training phase, where multiple decision trees are constructed using a random subset of training data as well as a random feature subset. In the second phase, a random feature set is selected for splitting at each node of every decision tree. The randomness in the feature selection ensures the diversity of features being selected across the forest. The final phase is the classification phase, where all the decision trees in the random forest algorithm vote on the label of the class. The label with the highest votes is chosen as the final classification output. RF is less prone to overfitting, and works well with data sets with missing data as well as high dimensional data. RF could be very computationally heavy and struggles to find meaningful patterns in a noisy data set [77, 78].

The development of the ML model was divided into two phases.

- **Phase 1:** To select the best performing ML algorithm, which was trained using open source databases on the internet.

- **Phase 2:** To adapt the model to the taring data generated by the Movesense ECG and HR monitor.

The first data set used for training the ML model is the MIT-BIH arrhythmia data set [79]. The data set consists of normal ECG signals and signals from cases affected by different heart arrhythmia conditions. The performance of CNN, KNN, and RF algorithms is discussed in the following section.

Implementation of Convolutional Neural Networks (CNN)

After several iterations, the final CNN developed was constructed using 8 convolutional layers and 2 dense layers with several maxpooling and dropout layers in between.

Figure 4.14 shows the performance of the CNN on the MIT-BIH data set. The CNN was able to achieve an accuracy of 98.21% after 250 epochs. One epochs is defined that the entire dataset has passed through the learning algorithm once during the taring phase.

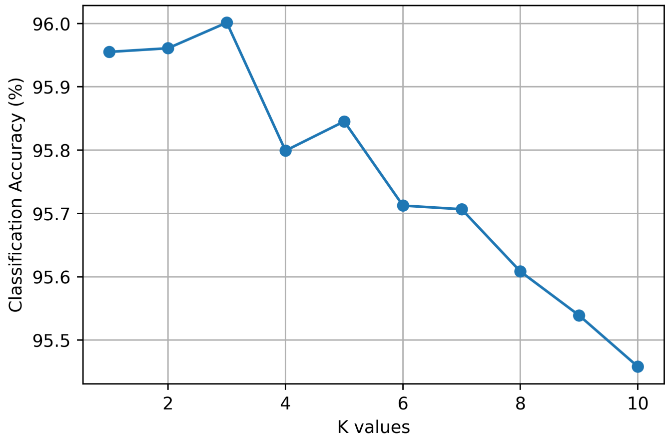


Figure 4.15: KNN classification performance of MIT-BIH arrhythmia data Set [78]

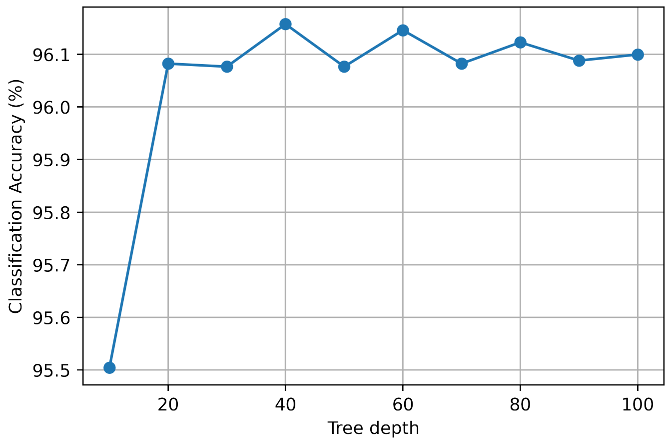


Figure 4.16: RF classification performance of MIT-BIH arrhythmia data Set [78]

Exposing the ML algorithm to the same dataset multiple time allows the model to learn different features of the dataset. As can be observed from the graph, during the first 100 epochs, the validation accuracy is more unstable. After approximately 150 epochs, the validation accuracy improves very slightly after each iteration of the gradient descent. Gradient decent is used for optimising of different weights used by the training algorithm. The idea is to find the optimal values for each of the weights so that the overall accuracy of the model increases.

Implementation of k-Nearest Neighbors (KNN)

Figure 4.15 shows the classification accuracy performance of the KNN on the MIT-BIH data set. The selected range of K is between 1 and 1o. As can be observed, the KNN algorithm has achieved the highest rate of 96% accuracy at K=3. The accuracy of the KNN rises until K=3, then drops as the value of K increases to 10.

Implementation of Random Forest (RF)

Figure 4.16 shows the classification accuracy performance of the RF algorithm on the MIT-BIH data set. The algorithm was set to evaluate the performance over the depth of the decision tree between 10 and 100. As can be observed from the accuracy performance, the accuracy of the RF is the highest at 96.16% at tree depth level 40.

Algorithm	% Accuracy	Training Time (S)
RF	96.16%	614.4
CNN	98.21%	495.6
KNN	96%	289.1

Table 4.1: ML models performance comparison (based on: [78])

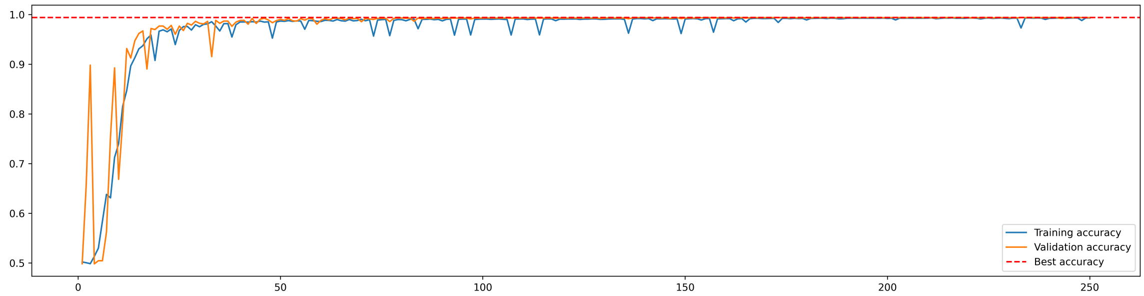


Figure 4.17: CNN classification performance of custom dataset [78]

Table 4.1 shows the classification accuracy outcome and training time comparison between RF, CNN, and KNN algorithms. As the number indicates, the CNN model has the highest classification accuracy than RF and KNN. Therefore, CNN was chosen as the preferred algorithm to be used in the deployed ML model.

After choosing the ML algorithm, it was time to train the CNN on the data generated by the Movesense ECG and HR monitor and the prototype device. In this case, the idea was to record the ECG of the people involved in this project. Therefore a data set of the ECG measurement was created by performing measurements on three volunteers. The total number of ECG measurements recorded was 20000 samples. When the data was first analyzed, it was evident that almost all the measurements were without any heart issues. This would have been a big challenge since the trained model would have been biased in its performance. Therefore, it was decided to develop a hybrid data set comprising 20000 entries, 10000 from the good ECG sample, and 10000 from abnormal cases from MIT-BIH and PTB-XL datasets. PTB-XL, is a large publicly available 12-led ECG dataset consisting of various 10 second long ECG recordings from 18885 patients cardiac arrhythmias and heart conditions [80]. This meant that the final data set combined 50% good data and 50% of abnormal data.

Figure 4.17 shows the training and performance accuracy graph of the CNN model as it can be observed from the graph the 99.37%, which is really good. For the first 50 epochs, the accuracy fluctuates, but after that, it stabilizes, and there is very little difference between training and validation accuracy.

To test the versatility of the developed CNN network, it was decided to test the performance of this network against the PTB-XL data set. PTB-XL data set contains 6528 samples of 12-LED ECG signal containing normal and abnormal data samples. For the taring of the CNN network, two data classes were chosen from the data set normal samples and samples with atrial fibrillation. This reduced the sample size for training from 6528 to 3587 samples.

Figure 4.18 shows the training and validation accuracy of the CNN network on the 12LED ECG signal data set. As can be seen from the graph, the performance accuracy of the

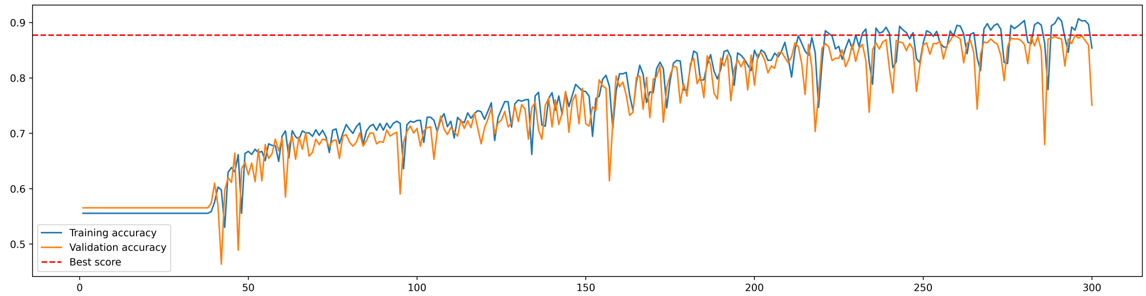


Figure 4.18: CNN classification performance of PBT-XL dataset [78]

Data Set	% Accuracy	Training Time (S)
MIT-BIH	98.21%	496.6
Custom	99.42%	488.2
PTB-XL	88.86%	176.6

Table 4.2: CNN models performance comparison (based on: [78])

CNN network is worse than the previous two data sets, with just 87.75%. The reason for this could be the low sample size of the data to train the network.

Table 4.2 shows the performance comparison of the CNN network when trained on three different data sets. As seen from the table, the performance of CNN in the custom data set is the highest. Therefore, to put the CNN for further testing, it was decided to make one last test with a completely different data set. The hypothesis here was that the custom data set used during the training of CNN was balanced, meaning the division between the normal and abnormal signals was 50-50. This balanced distribution is very unlikely to occur in a real-world scenario. Therefore, to replicate a realistic scenario, a new data set was created with 4000 abnormal signals and 10000 normal signals. This data set was not used for training the CNN but was used for just validation of the CNN. The accuracy of the CNN network for this new data set was recorded at 99.04%. This accuracy is very high but is a result of not having true abnormal data being recorded by the prototype end device. Therefore, to further improve the performance of the CNN network, gathering real ECG data from CVD patients and training the CNN model once again is essential.

4.5 Feedback from Clinical side

In this chapter, there were three different parts on which the feedback was collected. The feedback for the web portal, Android app, and ML Algorithm is described below:

WebPortal: heartrater.live

The overall feedback around the usability of the web portal was very positive. The designed portal was found to be very simple to use and navigate, and all the relevant information was easily accessible or could be found without much effort. The overall design of the web portal was found intuitive to navigate, well designed, and included relevant features.

It was also highlighted that the functionality offered by the portal was fundamental, but if it was included as a part of a complete system, it made sense. The presentation of realtime moving ECG was found cool, but it was pointed out that the ECG diagnosis is done on a still frame of ECG. This led to the addition of a pause button on the ECG portal which added the functionality of pausing the moving realtime ECG graph.

Android app: eMed

The Android app eMed was evaluated by performing a Heuristic Evaluation in which the usability of the user interface is measured by using the rule of thumb. Another tool used for testing the usability was using the protocol Think-Aloud, where data is gathered from the number of participants by asking a series of questions while using the prototype.

The overall feedback given for the app was good, and has the potential to serve as an MVP. Some of the feedback includes ease of use, a minimalist design, and intuitive navigation is present. Some drawbacks of the desired app include a need for a startup guide, error prevention, and error messages. The app has some unnecessary features, such as the total number of steps in a year. This information is relatively useless for the end user.

Machine Learning Algorithm:

Even though the ML algorithm can identify the abnormal ECG pattern from the regular ECG pattern, it still lacks the functionality to detect and identify between different CVD. In future versions of the ML models, it is very critical to use a rich dataset where different ECG patterns are recorded.

5 Network tests

The chapter Network Tests focuses on performing an experimental evaluation of the Cellular Internet of Things (C-IoT) technologies in different physical environments under different radio conditions. The experiments are designed to calculate the network Key Performance Indicator (KPI), such as End-to-End (E2E) latency, bitrate, and Packet Drop (PD). The rest of the chapter is divided into describing the experimental test setup and the results from C-IoT KPI testing in indoor, deep indoor, outdoor, deep-outdoor, and in roaming network conditions. Furthermore, the chapter includes evaluating the capacity and the network KPIs of a single eNB to support several simultaneous IoT devices. The chapter also includes performing tests to calculate the energy consumption of two C-IoT communication modules by sending TCP and UPD packets to a remote server. Finally, the chapter includes tests conducted to explore the use of C-IoT in applications involving continuous monitoring of patients under emergency transportation.

5.1 C-IoT test setup and Network KPI

5.1.1 Related Work:

There have been a number studies where authors have tired to evaluate the simulated as well as experimental performance of coverage, capacity, Quality of Service (QoS) offered by C-IoT technologies. Radheshyam *et al.* [81] experimentally shows that in case of NB-IoT the cell boundary is 800 meters, before the end device connects to another eNB to send data. Samir *et al.* [82] performance a simulation study where the LTE-M network performance is thoroughly evaluated for Machine Type Communication (MTC) devices and highlighed that the LTE-M technology is not ready yet to satisfy QoS the requirements of future IoT applications. Hassan *et al.* [83] performed coverage analysis of NB-IoT within 700 meters radius from the eNB to evaluate the coverage in outdoor, indoor, and underground locations. The tests results reveled that NB-IoT can provide connectivity to devices up to 400 meters underground. Shankar *et al.* [84] evaluates the QoS performance of commercially deployed NB-IoT networks in Belgium using Ublox SaRa-N211 module.

Based on the obtained results, they managed to achieve 17 Kbps maximum bitrate, less than 5 seconds of latency in both DL and UL communication. Massimo *et al.* [85] performance a comparison study of LoRaWAN and NB-IoT involving energy efficiency, QoS, and coverage for an industrial application. In their findings they discovered that NB-IoT energy consumption is 10X higher than that of LoRaWAN, at the same time NB-IoT is much more reliable and provides highest QoS. Ahmed *et al.* [86] evalualtes performance of NB-IoT under different deployment modes and found that NB-IoT performs the best in guard band deployment in single-tone configuration. Krzysztof *et al.* [87] experimentally evaluates the signal strength and quality of NB-IoT and compares it's performance with LoRaWAN in marine environment. The results show that NB-IoT signal penetration is higher than that of LoRaWAN. Sebastian *et al.* [88] provides a performance comparison between LPWAN and C-IoT networks. The tests revels that C-IoT has lowest observed packet drop, latency, and range than that of other LPWAN technologies. Andre *et al.* [89] highlights the difference between modeled and experimental battery life estimations for NB-IoT and LTE-M and show that the battery life performance of devices using LTE-M and NB-IoT can reach upto 10 years as highlighted by 3GPP by optimising application specific and network specific parameters. Sebastian *et al.* [90] performance emulated as well as experimental measurements of NB-IoT networks. The performance study includes coverage, latency, and protocol consistency. The tests were performed on two MNO Vodafone and T-Mobile. The average throughput in DL direction was lower than UL direction, and the average RTT recorded was 878.94 ms & 1055,67 ms for Vodafone & T-Mobile respectively.

Although the existing research gives a good starting point in understanding the performance of C-IoT networks, most of the studies include using either simulations, use of only one MNO, or only one test environment, or just using one of the C-IoT technology. Therefore, in order to understand the true nature of these technology in a physical heterogeneous network deployment it was decided to conduct a through experiemtnal evaluation of both NB-IoT and LTE-M during this project. The purpose behind testing the C-IoT networks was to identify the performance of NB-IoT and LTE-M in a realistic deployment scenario for continuous monitoring applications. Therefore, it was decided to test both NB-IoT and LTE-M from an E2E perspective in different deployment scenarios. The end goal was to generate some network performance KPI, which can be used to benchmark the behavior of these technologies across different applications. In order to keep the test setup and the tests as generic as possible, standard off-the-shelf components were used. The tests evaluated generic network KPI such as latency, bitrate, and PD in various radio conditions. Different MNO networks can have varying E2E network performances; therefore, all the tests were conducted using Denmark's two main radio network operators. Both of these operators, at the time of testing, had nationwide NB-IoT and LTE-M coverage. The C-IoT device, test setup architecture, and critical KPI for the tests are defined below:

Test End Device:

Figure 5.1 shows the overview of the test device used for carrying out all the network KPI tests. In order to keep the results obtained as generic as possible, it was decided to use a commercially available off-the-shelf C-IoT device. PYCOM FIPY is a commercially available test device that was updated to the latest firmware when testing the network KPI [91]. PYCOM FIPY device was chosen for the following reasons:

1. PYCOM FIPY supports both the C-IoT technologies (LTE-M and NB-IoT), which makes it easier to test both technologies using the same hardware. This meant that

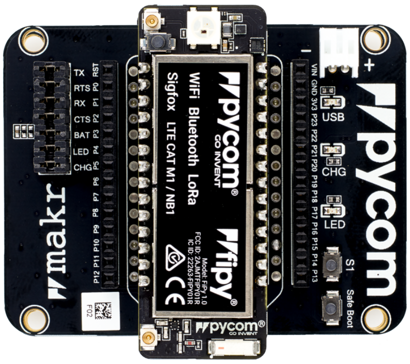


Figure 5.1: Test End-Device

it was safe to assume that the obtained results would only differ because of changes in the network. During the tests, it was also ensured that the same device and it's peripheral components were used to perform all the tests (i.e. one device for one C-IoT technology and one MNO). This helped us assume even fewer variations in the end results caused by the hardware.

2. The board's design is very compact, making it possible to test multiple MNO and C-IoT technologies from the exact location simultaneously. This was especially important because cell loading and availability of the network resources could affect the calculation of the KPI making the test results less comparable.

3. PYCOM FIPY also supported using a Micro-SD card to store all the results locally on the device. This made it possible to conduct tests without requiring additional computers during the tests.

Test Network Architecture

Figure 5.2 shows the complete E2E test network architecture. All the test scenarios use the same architecture to conduct the KPI tests. Once powered, the device is connected to Operator One (OP1) or Operator Two (OP2). The choice of the operator is defined in the setup sequence of the firmware flashed onto the device at the beginning of the test. Depending upon if it is an NB-IoT or LTE-M network test, the test device connects to the appropriate Radio Access Technology (RAT). Once connected, the device sends UDP packets of different payload sizes to the application server. In order to keep the network architecture generic, the application server is hosted on a cloud service provider's data center located in Frankfurt, Germany, which is approximately 850km to 900km away, depending on the test locations. This architecture allows data to be sent over the public internet, where it is assumed that all the traffic is handled in a best-effort manner, and it can also be safe to assume there are no particular priorities assigned to traffic coming from different MNO.

Network KPI calculations

Figure 5.3 describes the overall setup procedure and highlights how the network KPI

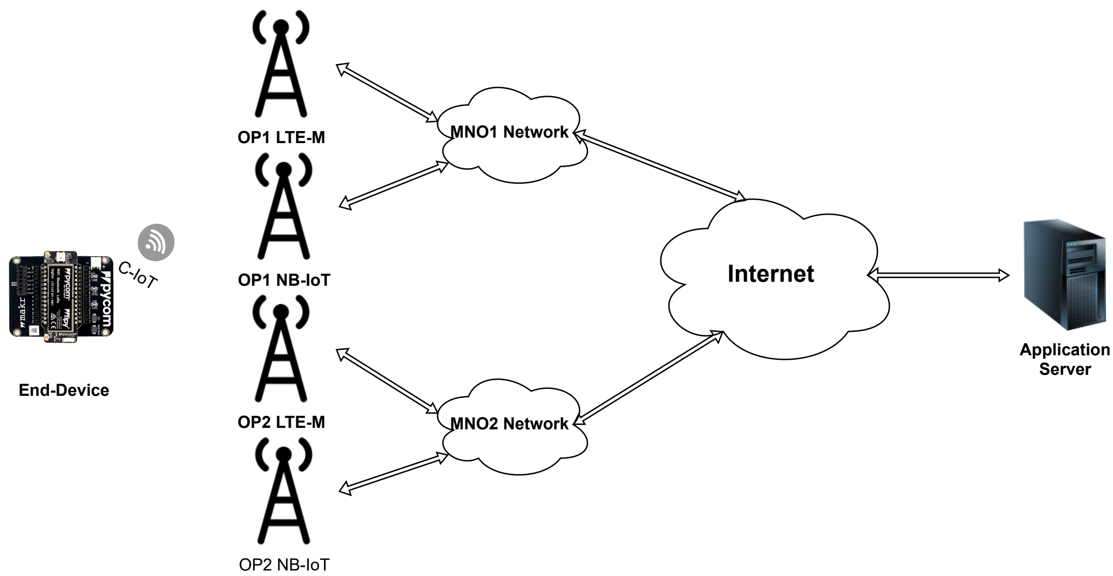


Figure 5.2: Test Network Architecture

were calculated. Once the device boots up, based on the choice of operator and the technology to be tested, follows the setup procedure. The operator and the technology choice are defined in the network attach parameters in the firmware. Once the device is connected to the network, it requests the Packet Data Protocol (PDP) context using the appropriate Access Point Name (APN). In order to keep the setup as generic as possible, both device requests for generic public APNs allowed by the two MNO. This ensures that the traffic sent using both these devices follows the default network settings applied by the MNO. Once the IP address is allocated to the device, the device opens a UDP socket and sends data to the application server. Once the data is sent, the device waits for the acknowledgment packet to arrive from the server. The network KPI are calculated using the following logic:

- **Latency:** When the device sends the UL UDP packet, it keeps an entry of the timestamp at which the UL packet was sent. When the application server receives this packet, it sends a DL acknowledgment packet to the device. Once this packet arrives at the device, it takes the arrival timestamp of this packet. Once the device has both sent and arrival timestamps, the device subtracts these two values to calculate the E2E latency experienced by the packet.

- **Bitrate:** Similar to the end device, the server also keeps an arrival rate between the two consecutive incoming packets from the same end device and calculates bitrate in Kilobits per Second (Kbps).

- **Packet Drop (PD):** During these UL and DL packets, if for some reason the DL packet does not reach the test device because either the UL packet did not reach the server or the DL acknowledgment packet was dropped. The end device considers the packet to be dropped after a predefined timeout in the firmware.

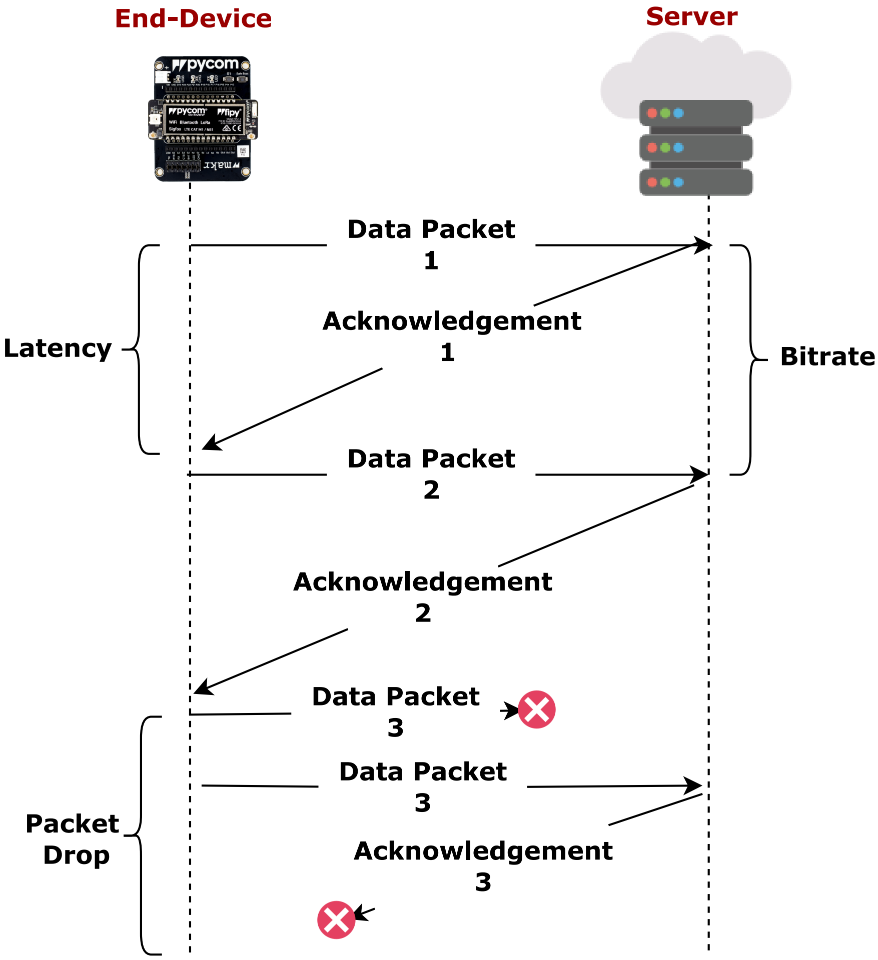


Figure 5.3: Experimental Setup and KPI calculation

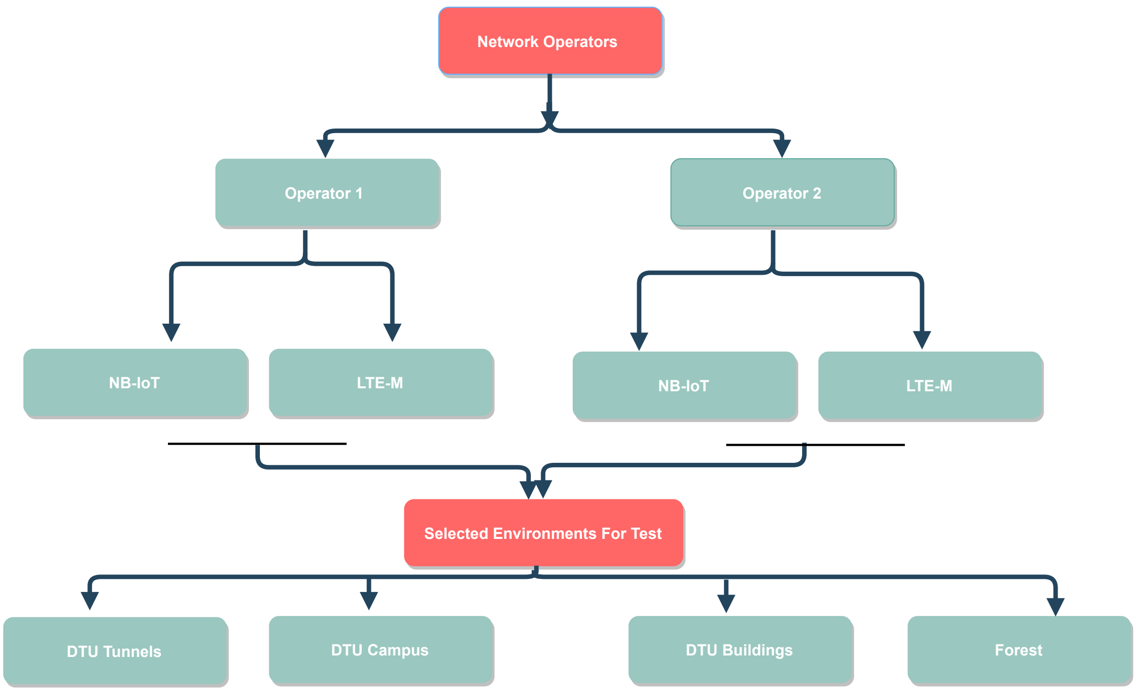


Figure 5.4: Network KPI Test Scenarios

5.2 Network tests

The network KPI testing involved simultaneously testing both Operators 1&2 in the same environment using both NB-IoT and LTEM-M RATs. This was especially important due to the dynamic nature of cellular communication technologies and the effects of network traffic patterns across 24 hours. Figure 5.4 shows the different network KPI test scenarios.

The network KPI test scenarios where the tests are performed were identified based on the active dialog with the medical professionals, clinical researchers and the CVD patients. The whole idea behind designing such a remote heart monitoring system was to monitor the patients' activity no matter where they go. Therefore, the requirements were narrowed down to four different environments and then further translated into realistic scenarios that closely resemble the challenges presented by each of these environments. The four test environments were indoor, deep indoor, outdoors, and remote outdoors.

The first test environment chosen was performing tests indoors. This could mean testing the network KPI when the patient wearing the device can be monitored in a confined space, e.g., a house, hospital, rehabilitation center, etc. Therefore, a regular Technical University of Denmark (DTU) office building was chosen to resemble such an environment.

The second test environment chosen to perform KPI testing was deep indoor. This could mean testing the network performance in environments such as house basements, parking places, etc. Therefore, to closely resemble such an environment underground tunnel system at the DTU was chosen. The underground tunnel system of DTU covers the entire university campus, and a part of this tunnel system was chosen to perform the experiments. The route was explicitly designed to start away from the eNB on campus and walk towards it; in other words, the device goes from an inferior coverage area to a better one.

The next tests were conducted in an outdoor environment. This could mean performing the network KPI testing on streets, parks, etc. It was decided to use different streets around the campus to carry out the tests. In order to keep the deep indoor and outdoor tests more relatable, it was decided to perform the outdoor tests following the same path as the deep indoor tests.

The final test environment chosen for network KPI testing was remote outdoor. This environment was supposed to capture network KPI performance in hiking trails, walking paths in the forest, etc. This was especially interesting given that most CVD patients are instructed to carry out outdoor exercises during the rehabilitation period. The closest environment where we could resemble such an environment was at Rude Skov (Rude Forest), which has several walking trails and is about 10km north of the university [92].

5.2.1 Indoor Test Scenario

In this section, the different tests intend to emulate the indoor environment in which the device could be used. Therefore, an office phasing the direction of the on-campus C-IoT eNB was chosen to replicate the scenario. The idea was to use the PYCOM FIPY test device to send UDP packets to the application server hosted online. The packets were sent using different payload sizes 128 Bytes, 256 Bytes, 512 Bytes, and 1024 Bytes. The tests followed the same experimental setup described in figure 5.3. The tests were kept running without any interruption for a total duration of five days prior to the Christmas

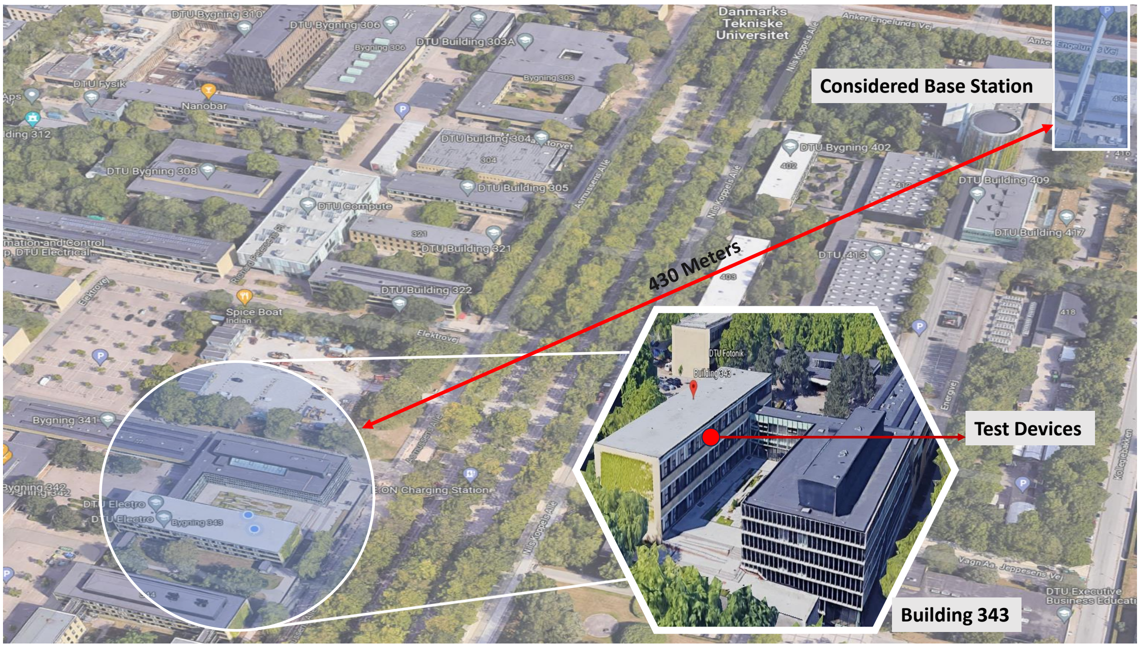


Figure 5.5: Indoor test location

holidays. The timeline was chosen under the assumption that there will be less activity on the campus and the tests will not be affected due to extreme traffic patterns experienced by the mobile networks. However, unfortunately, the LTE-M tests failed twice during this period, and NB-IoT failed once, and the experiment needed to be restarted.

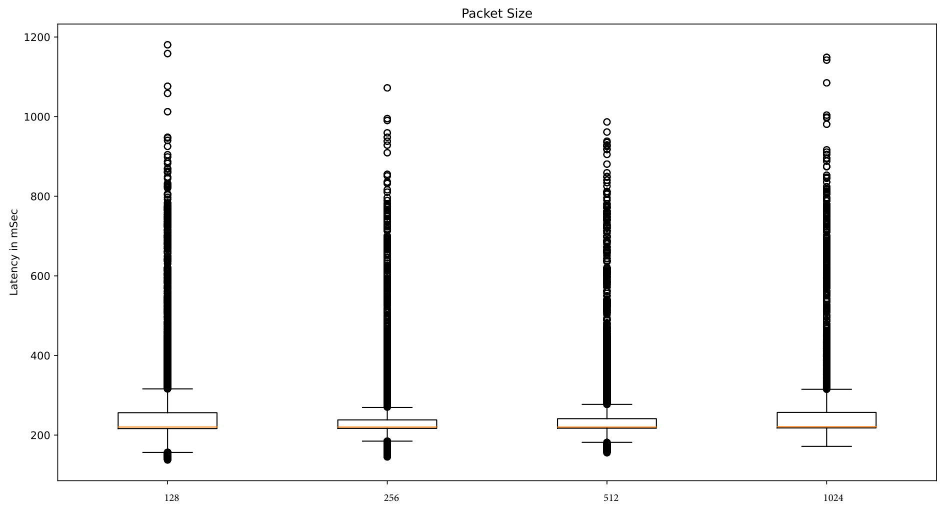
Table 5.1 describes the different test attempts and the number of packets sent in each attempt. The primary reason to have these tests run over multiple days was to try to evaluate the E2E performance of the C-IoT over an extended period and to check whether the performance varies during the peak vs. non-peak traffic hours experienced by the C-IoT cells.

Test attempts	No. of packets OP1 LTE-M	No. of packets OP1 NB-IoT	No. of packets OP2 LTE-M	No. of packets OP2 NB-IoT
1	20104	3504	20004	7580
2	51100	101912	50004	53204
3	100300	NA	100104	NA

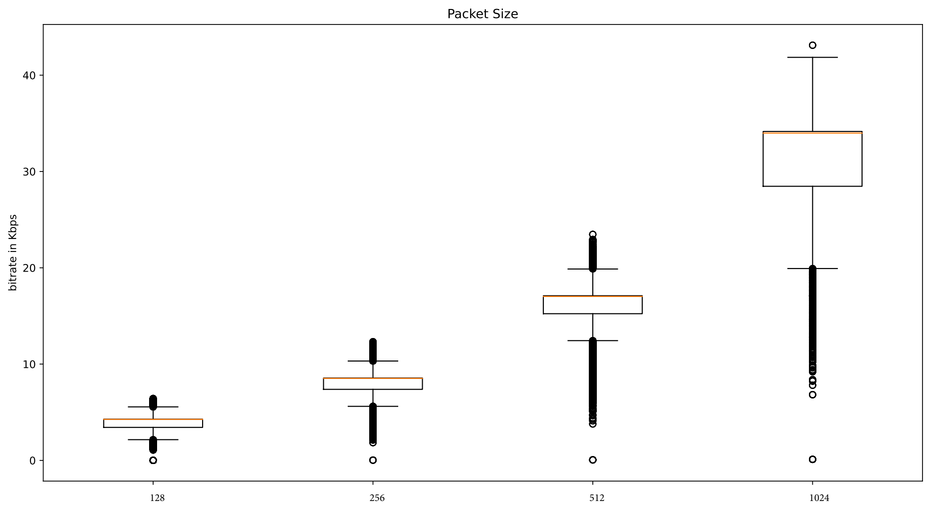
Table 5.1: Indoor test attempts

Figure 5.6 shows the LTE-M E2E latency and bitrate calculated over 171504 packets. In this case, the box plots are drawn over the raw data recorded by the device and the application server. Figure 5.6a shows the box plot for E2E latency observed by OP1 while performing the KPI testing using varying payload sizes. As can be observed from the plots, the median E2E latency value across all the payload sizes is very similar. The highest latency experienced during OP1 LTE-M KPI testing was 13.075 Seconds where, as the minimum latency observed by OP1 during the tests was 35.65 mSec.

Figure 5.6b shows the bitrate calculated during the same test across all the payload sizes. As can be seen from the graph, the recorded bitrate of the OP1 LTE-M network goes on increasing as the payload of the packet increases. This is an expected behavior, as



(a) OP1 LTE-M E2E Latency in indoor environment

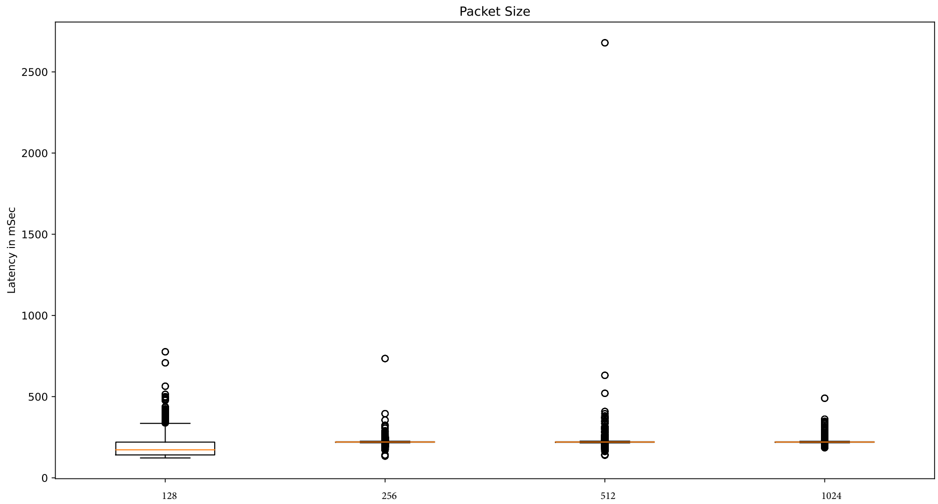


(b) OP1 LTE-M bitrate in indoor environment

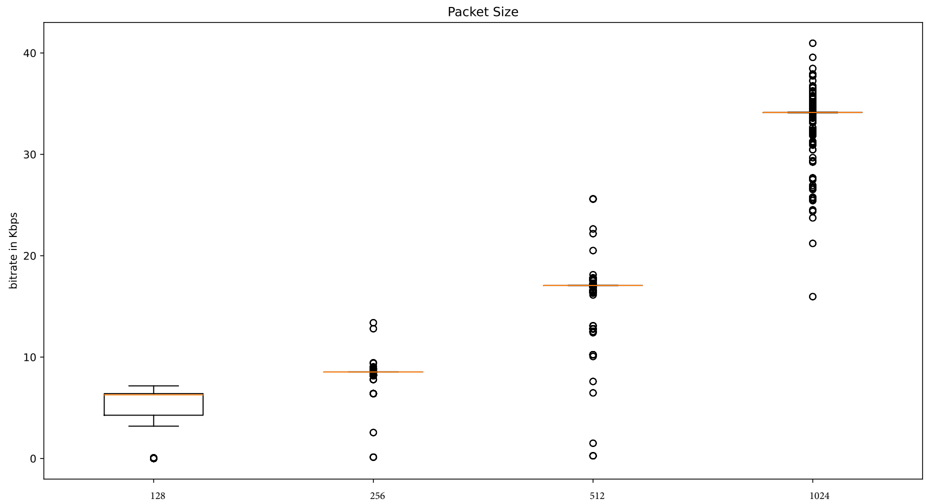
Figure 5.6: OP1 LTE-M KPI in indoor environment

the increase in payload meant more bits/second received at the application server. The highest bitrate recorded for the OP1 LTE-M network during the testing was 55.652 Kbps, whereas the lowest bitrate recorded during the test was 0.02 Kbps.

Figure 5.7 shows the OP2 LTE-M E2E latency and bitrate performance results. This performance was calculated over 170112 packets of varying payload. Figure 5.7a shows the box plot results for calculated E2E latency over different packet sizes. It can be seen from the plots that the calculated median latency for 128 Bytes packets is slightly lower in comparison with the rest of the payload sizes. The highest value of E2E latency recorded was 7.541 seconds, whereas the lowest recorded value was 50.99 mSec. Figure 5.7b



(a) OP2 LTE-M E2E Latency in indoor environment

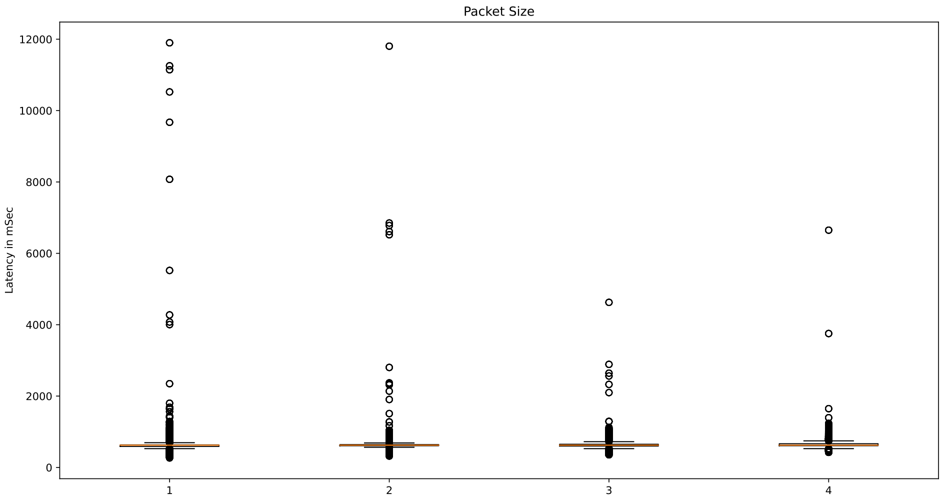


(b) OP2 LTE-M bitrate in indoor environment

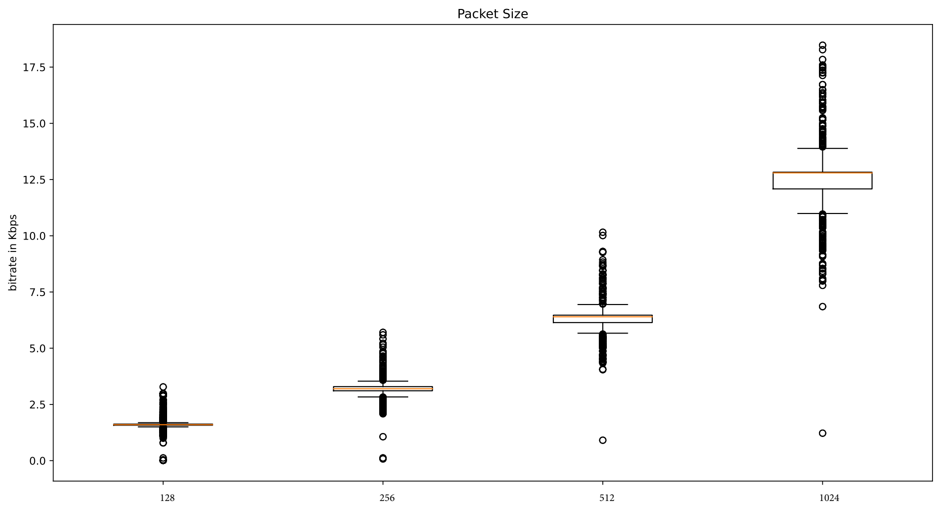
Figure 5.7: OP2 LTE-M KPI in indoor environment

shows the observed bitrate performance during the same test. Similar to the case for OP1, in the case of OP2, the median bitrate increases as the payload size increases. The highest calculated bitrate during the OP2 LTE-M was 42.89 Kbps, whereas the lowest value was 0.02 Kbps.

Figure 5.8 shows the OP1 NB-IoT KPI performance results in an indoor environment. The KPI performance was calculated over 105416 packets of varying payload sizes sent to the application server. Figure 5.8a shows the E2E latency experienced by NB-IoT packets throughout the test. As can be seen from the plots, the median latency recorded for all the packet sizes is is very similar. The highest E2E observed during the OP1 NB-IoT testing



(a) OP1 NB-IoT E2E Latency in indoor environment

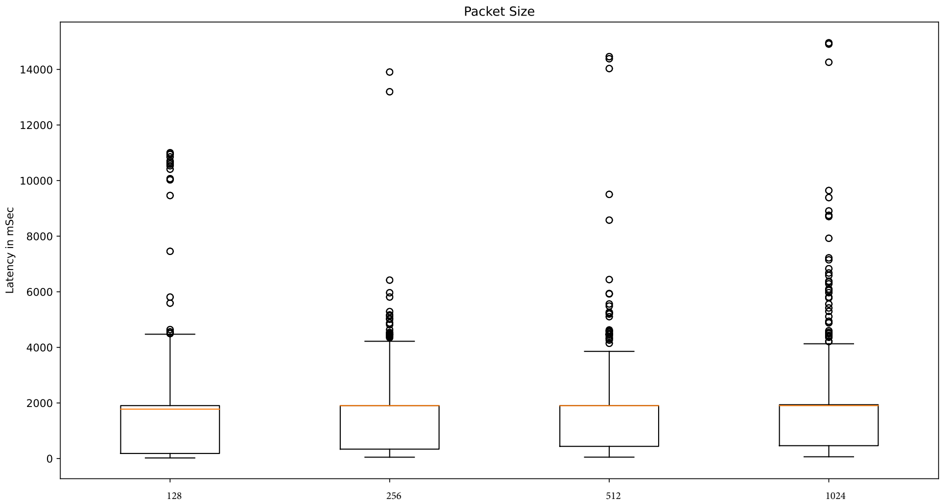


(b) OP1 NB-IoT bitrate in indoor environment

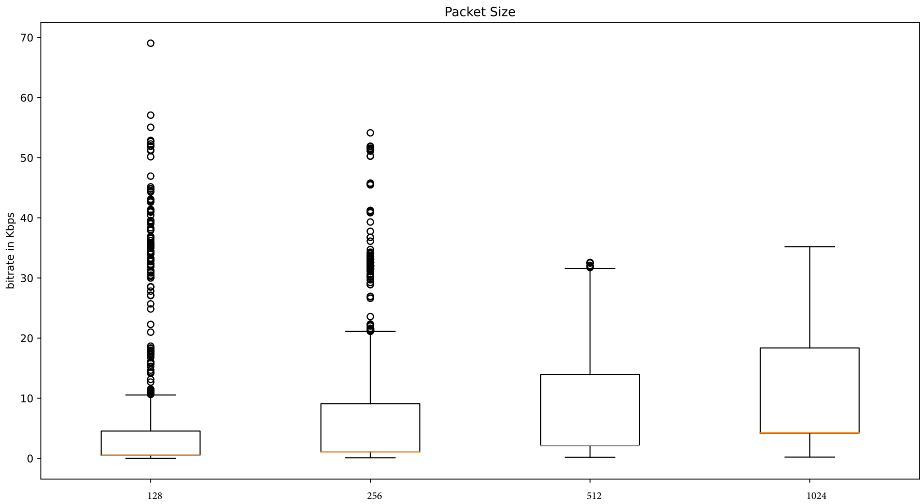
Figure 5.8: OP1 NB-IoT KPI in indoor environment

was 14.833 Seconds, whereas the lowest observed E2E latency was 28.93 mSec. Figure 5.8b shows the calculated bitrate during OP1 NB-IoT testing. Similar to the LTE-M testing for OP1, the NB-IoT bitrate increases as the payload size of the packet increases. The highest calculated in case OP1 NB-IoT was 19.229 Kbps, whereas the lowest calculated bitrate was 0.002 Kbps.

Figure 5.9 shows the box plots for the bitrate and E2E latency recorded during the KPI testing of OP2 NB-IoT network in an indoor environment. The KPI values were calculated over 60784 packets sent using varying payload sizes. Figure 5.9a shows the E2E latency observed for the OP2 NB-IoT network. As can be seen from the graphs, the median value



(a) OP2 NB-IoT E2E Latency in indoor environment



(b) OP2 NB-IoT bitrate in indoor environment

Figure 5.9: OP2 NB-IoT KPI in indoor environment

for observed E2E latency is similar across all the payloads. The highest recorded E2E latency value for OP2 NB-IoT was 14.979 Seconds, and the lowest recorded value was 22.07 mSec.

Similarly, Figure 5.7b shows the recorded bitrate during the same KPI testing. As seen from the plots, in the case of OP2 NB-IoT, the median value during all the tests shows a very slight improvement as the payload size of the packet increases. The highest bitrate value recorded during the KPI testing was 70 Kbps, whereas the lowest recorded value was 0.001 Kbps.

The table 5.2 summarises all the test results. The summary table, along with Minimum

Technology	E2E Latency (mSec)		Bitrate (Kbps)		PD	Total Packets Sent	Drop Rate
	Min	Max	Min	Max			
OP1 LTE-M	35.65	13075	0.02	55.652	225	171504	0.13%
OP1 NB-IoT	28.93	14833.48	0.002	19.229	593	105416	0.56%
OP2 LTE-M	50.99	7541.08	0.02	42.89	75	170112	0.04%
OP2 NB-IoT	22.07	14979	0.001	70	2138	60784	3.51%

Table 5.2: Summery of KPI measurements for indoor environment

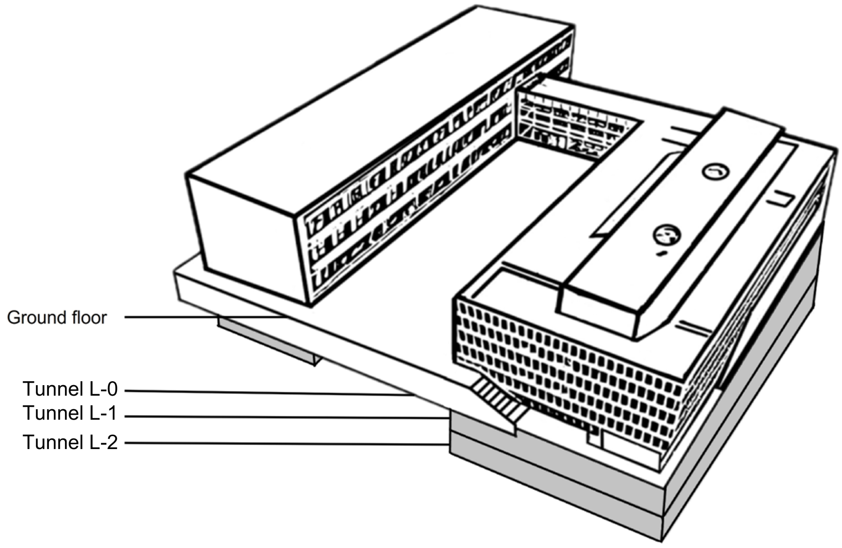


Figure 5.10: DTU deep indoor KPI test location

and Maximum values for E2E latency and bitrate, also summarises the PD during each test. In the case of OP1, the LTE-M and NB-IoT PD rates are very low, 0.13% and 0.04% , respectively. On the other hand, the PD ratio for OP2 LTE-M is 0.56% but has the highest drop rate ratio of 3.51% for the NB-IoT.

Based on the KPI tests performed during the five-day interval using OP1 and OP2 C-IoT network, it was observed that the payload size had very little to no effect on the overall E2E latency performance values. Another noteworthy observation was that both OP1 and OP2 had comparable network performance when it came to LTE-M networks. Unfortunately, it was a completely different story in the case of the NB-IoT network. The OP1 NB-IoT network offered superior network performers than OP2 in the indoor environment KPI tests. A possible hypothesis for the observed performance gap could be caused by the deployment configuration (in-band or guard band) of NB-IoT. This hypothesis could not be confirmed or further analyzed due to a lack of knowledge of the OP2's network deployment settings.

5.2.2 Deep Indoor Test Scenario

Figure 5.11 shows the path which was taken while testing the Basement and Outdoor network test scenario. As can be seen from the graph, the network KPI test was conducted on 17 different locations, which were 100 meters apart from each other. This path was

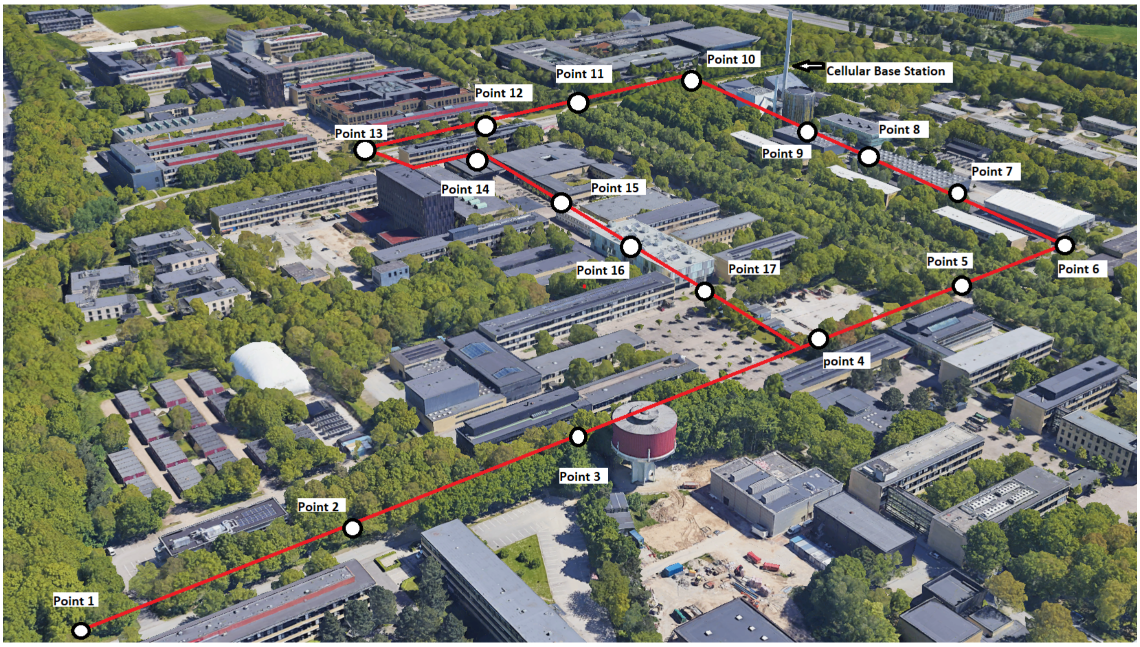


Figure 5.11: DTU Tunnel System Overview and KPI Test Path

specifically designed to test the network KPI further from the test eNB on campus. Figure 5.10 shows the different underground levels of the measurement path. Point 1 to point 9 were located in the tunnel level L-1 whereas point 10 to point 17 were located in tunnel level L-2. Before conducting the tests, it was assumed that only the on-campus eNB would be used during all the tests. This assumption was based on the previous study conducted that highlighted the maximum distance before a NB-IoT device changes the connected cell was 800 meters [81]. The distance between the furthest measurement point and the considered eNB was 525 meters (measured using google maps). In order to verify this assumption, especially in the deep indoor scenario, the cell id of the connected C-IoT device was also recorded during each test.

Figure 5.12 shows the median performance values of calculated E2E latency observed by both the operators for LTE-M and NB-IoT. In the graph, the X-axis represents the different points at which the KPI tests were performed, while Y-axis represents the latency calculated for both the RATs on a given test point.

As observed in the first graph, the LTE-M performance of the OP1 is very stable and consistent across the different points compared to the OP2. The median E2E latency calculated for LTE-M OP1 is 245.41 mSec, whereas, for OP2, it is 215.11 mSec. As one can see from the graph at point 4, both OP1 and OP2 have no coverage resulting in an empty test. Similarly, in the case of OP2, there is no coverage at points 3, 5, 11, 17, and 18. At point 2, OP1 has experienced the highest E2E latency of 563 mSec, whereas, at point 1, OP2 has experienced the highest E2E latency of 628.16 mSec. As can be observed from the graphs, even though OP2 had lower median latency values across the different KPI measurement points, OP1 offered a very stable network performance overall.

Regarding NB-IoT performance measured in the case of both operators, the story is very similar to that of LTE-M; OP1 had overall stable and consistent performance across all the points compared to OP2. In the case of OP1, the median E2E latency observed during NB-IoT testing was 625.76 mSec, whereas, for OP2, that number was around 848.47

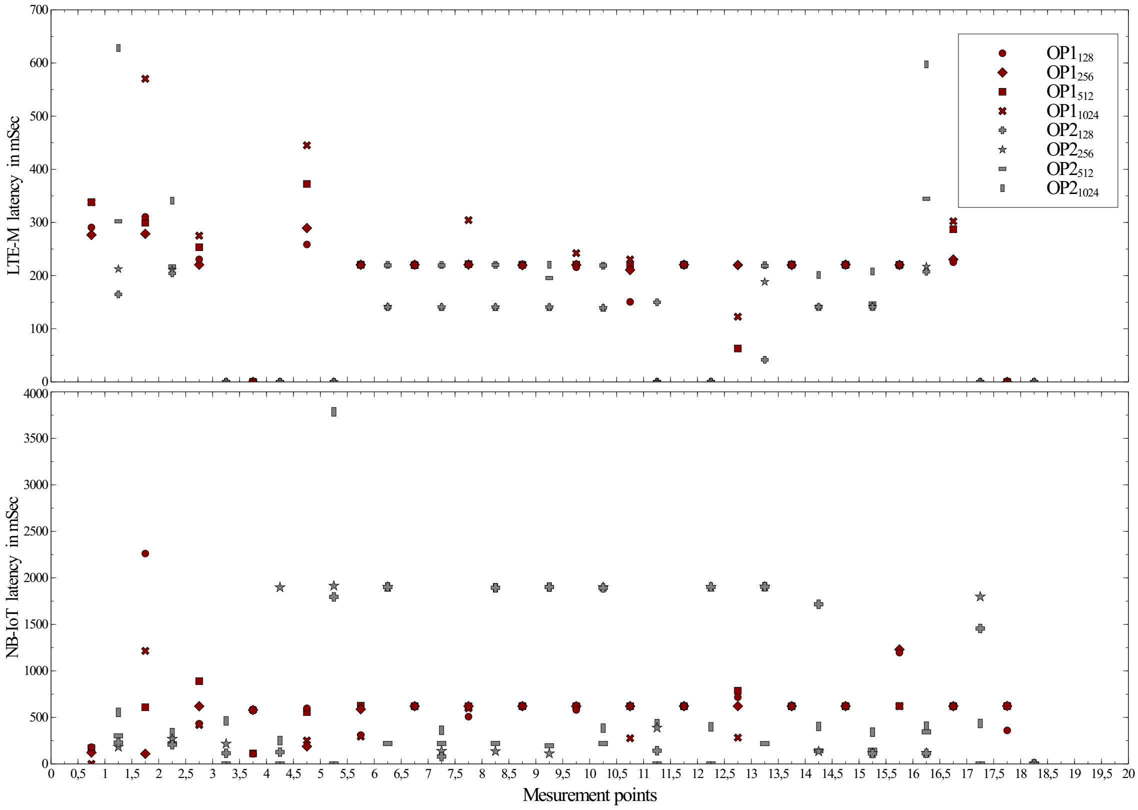


Figure 5.12: E2E Latency at all points deep indoor (based on [93])

mSec across all the measured points. OP1 experienced the highest median latency of 2258.12 mSec where, whereas OP2 experienced the highest median latency of 3785.04 mSec.

Figure 5.13 shows the recorded bitrate in Kbps for LTE-M and NB-IoT across all deep indoor points. The X-axis represents the different points where the tests were conducted, and the Y-axis represents the bitrate in Kbps. In the case of LTE-M, just like in the E2E latency tests, there is no observed coverage at point 4 for both the operators, and points 3, 5, 11, 17, and 18 had no OP2 coverage. At point 1, OP1 has the highest median bitrate of 63.589 Kbps, whereas, at point 15, OP2 has the highest median bitrate of 56.8 Kbps. In the case of the lowest median bitrate, OP1 observed 3.17 Kbps at point 2, whereas OP2 observed 3.25 Kbps at point 1.

In the case of NB-IoT highest and lowest bitrates were observed by OP1 at points 13 and point 2 with 29.82 Kbps and 0.62 Kbps, respectively. Similarly, for OP2, the highest median NB-IoT bitrate was observed at point 15 at 22.58 Kbps, whereas the lowest bitrate was observed at point 2 at 0.58 Kbps.

Figure 5.14 shows the recorded PD in UL and DL direction by OP1 on LTE-M and NB-IoT. As shown in the case of LTE-M, OP1 has the highest DL PD of 3045 packets at point 1, and the lowest PD of 2 packets is recorded at point 6. Similarly, the highest UL PD of 124 packets is recorded at point 2, and the lowest UL PD of 1 packet is observed at point 16.

Regarding NB-IoT, in the case of OP1, the highest DL PD of 1027 packets is recorded at point 6, whereas the lowest DL PD of 5 packets is recorded at point 11. The highest UL PD in the case of NB-IoT was recorded at point 18 with 38 packets, and the lowest UL

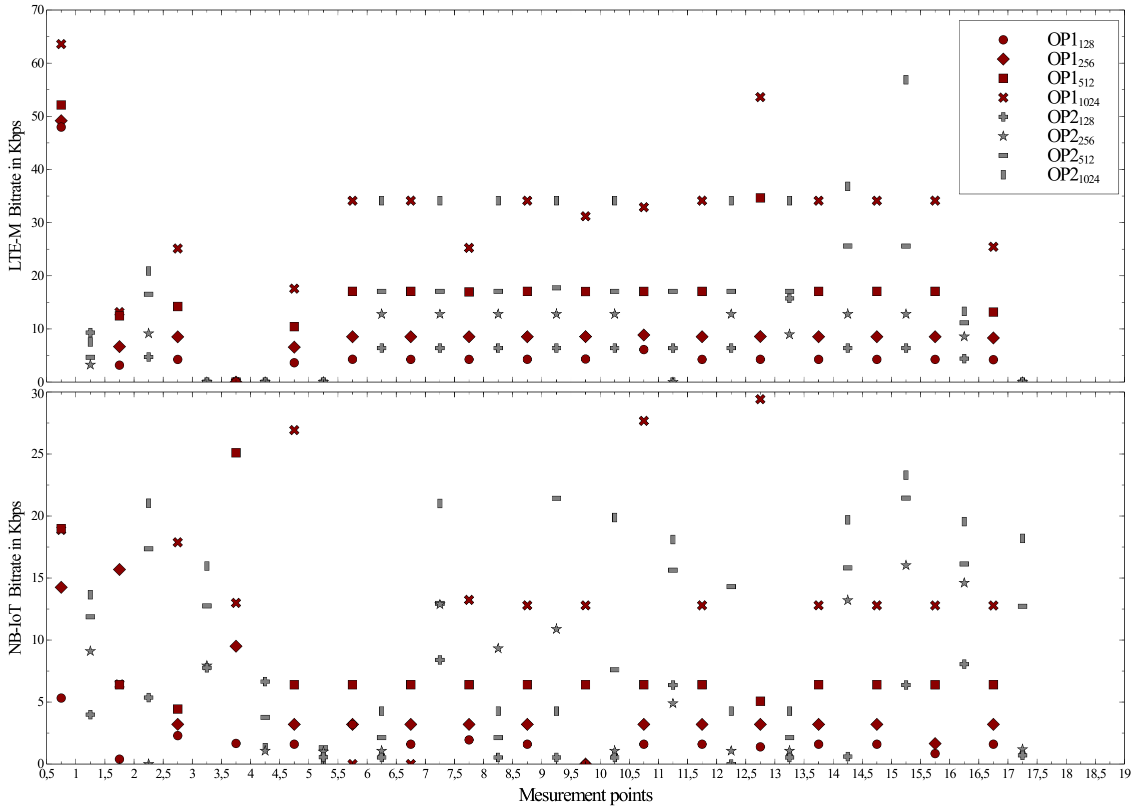


Figure 5.13: Bitrate at all points deep indoor (based on [93])

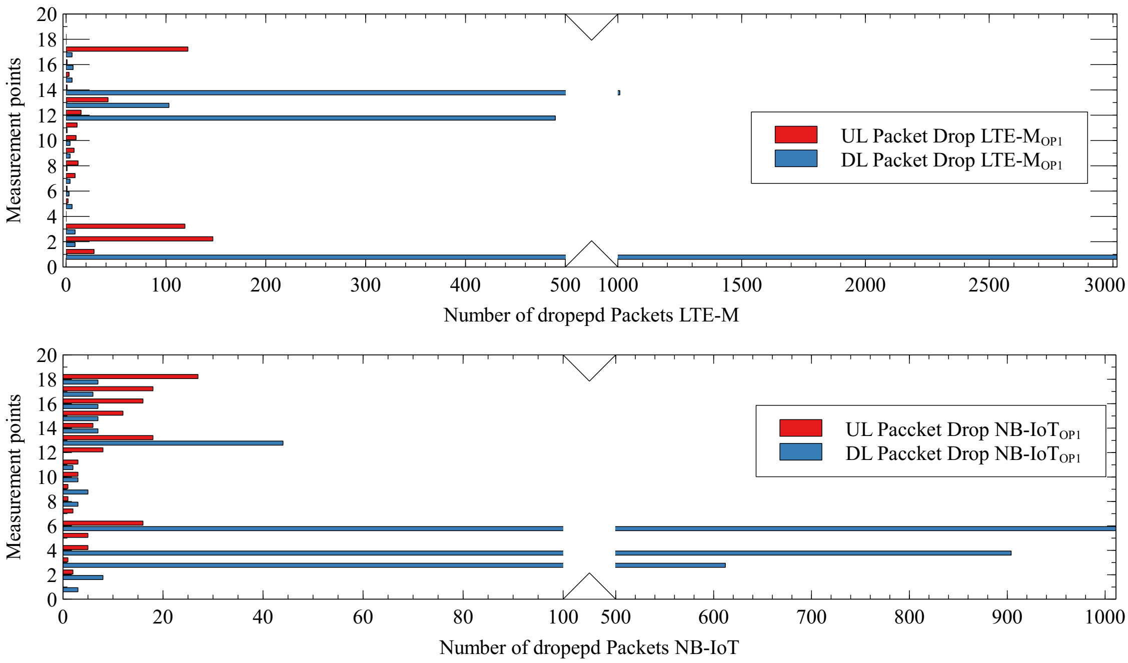


Figure 5.14: PD by OP1 at all points deep indoor [93]

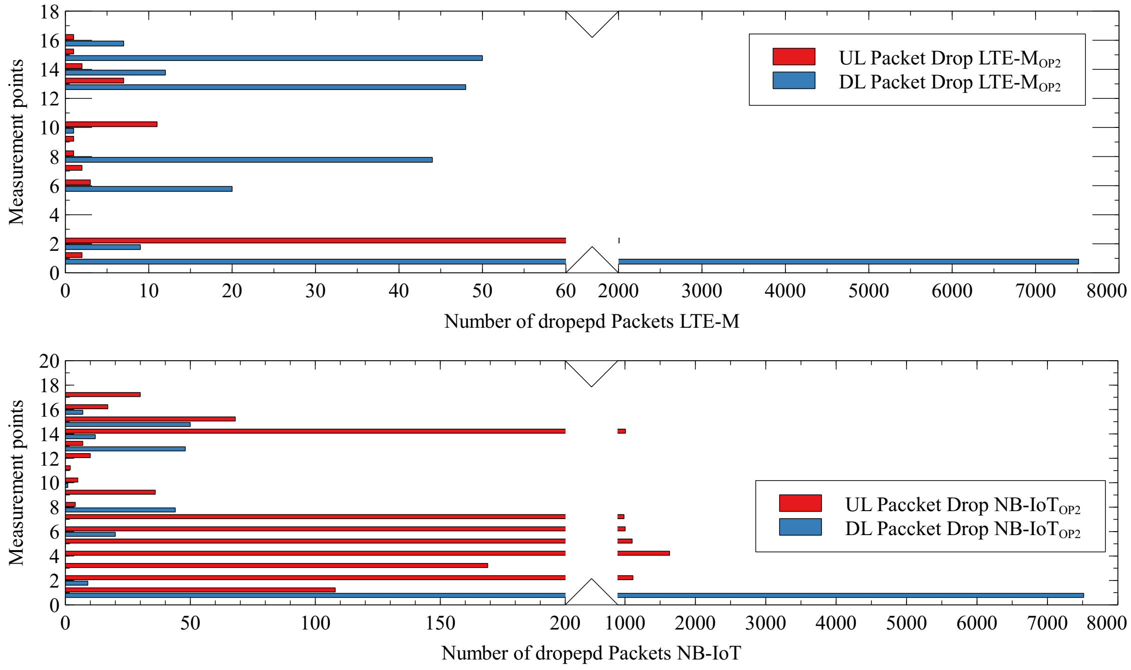


Figure 5.15: PD by OP2 at all points deep indoor [93]

PD of 2 packets was recorded at points 3, 8, and 9.

Figure 5.15 shows the recorded PD for OP2 during the deep indoor testing for both the C-IoT technologies. In the case of LTE-M for OP2, the highest DL PD of 7518 packets was recorded at point 1, whereas the lowest PD of 0 packets was recorded at point 9. Similarly, the highest LTE-M UL PD for OP2 was recorded at point 2 with a drop count of 2008, and the lowest UL PD was at points 8, 9, 15, and 16 with a packet count of 4.

In the case of NB-IoT deep indoor tests for OP2, the highest DL and UL PD was recorded at points 1 and 4 with 7516 and 1658 packets, respectively. Similarly, the lowest DL PD was recorded at points 3, 4, 5, 7, 9, 11, and 17 with no dropped packets. The lowest UL PD was recorded at point 11, with 3 UL packets dropped.

The KPI performance of OP1 in deep indoor conditions was very stable and consistent. At point 4, both OP1 and OP2 observed no LTE-M coverage. After further analysis, it was discovered that points 3, 4, and 5 were directly under a parking lot and concrete roads with multiple buildings blocking the LoS towards the eNB. Furthermore, Figure5.16 shows the deep indoor environment where the KPI testing was conducted. The figure shows that the test environment is constructed using heavy concrete walls, which can attenuate the cellular signal significantly. Not only that, the test environment was surrounded by water pipes, high-voltage electrical cables, and fiber optic cables, which all could contributed in reducing Signal-to-Noise Ratio (SNR).

After looking into the metadata from different tests, it was identified that the test device was connecting to two eNB in the case of OP1 and three different eNB in the case of OP2. Therefore, the performance differences observed during the test might have been due to the uneven availability of the RAN resources. The information about the location of different eNB during the different tests was unavailable from the MNO, hence this theory

Figure 5.16: Deep indoor environment Hardware Test Setup

could not be further tested and verified.

In the case of the PD tests, there was an evident trend observed in the DL packet loss being higher than the UL packet loss. A possible hypothesis for this trend could be:

1. Possible In-band deployment configuration of NB-IoT and LTE-M having to share the same resources as regular LTE UEs. This could lead to radio resource congestion during peak hours and lower performance.

2. Network congestion could lead to long delays, which were higher than the timeout time set in the firmware.

3. DL packet loss due to use of Network Address Translator (NAT). Both operators OP1 and OP2 allotted Privet IP addresses to the test devices.

5.2.3 Outdoor Test Environment

The KPI testing in the outdoor environment is conducted on the path highlighted in Figure5.11 around the university campus. The same path is used for conducting the deep indoor environment tests, and the test location of the points coinside. Figure 5.17 shows the hardware setup used for the KPI testing in an outdoor test environment. This setup is the same as the one used in the deep indoor environment tests. All other aspects of the tests were kept identical. Therefore apart from the test location being in the outdoor environment, there were no changes to the overall E2E test setup.

Figure 5.18 shows the E2E latency recorded while testing both LTE-M and NB-IoT using both OP1 and OP2. The LTE-M measurements indicate that the tests performed using OP1 have consistent E2E latency values. The median E2E latency observed in the case of

Figure 5.17: outdoor environment Hardware Test Setup

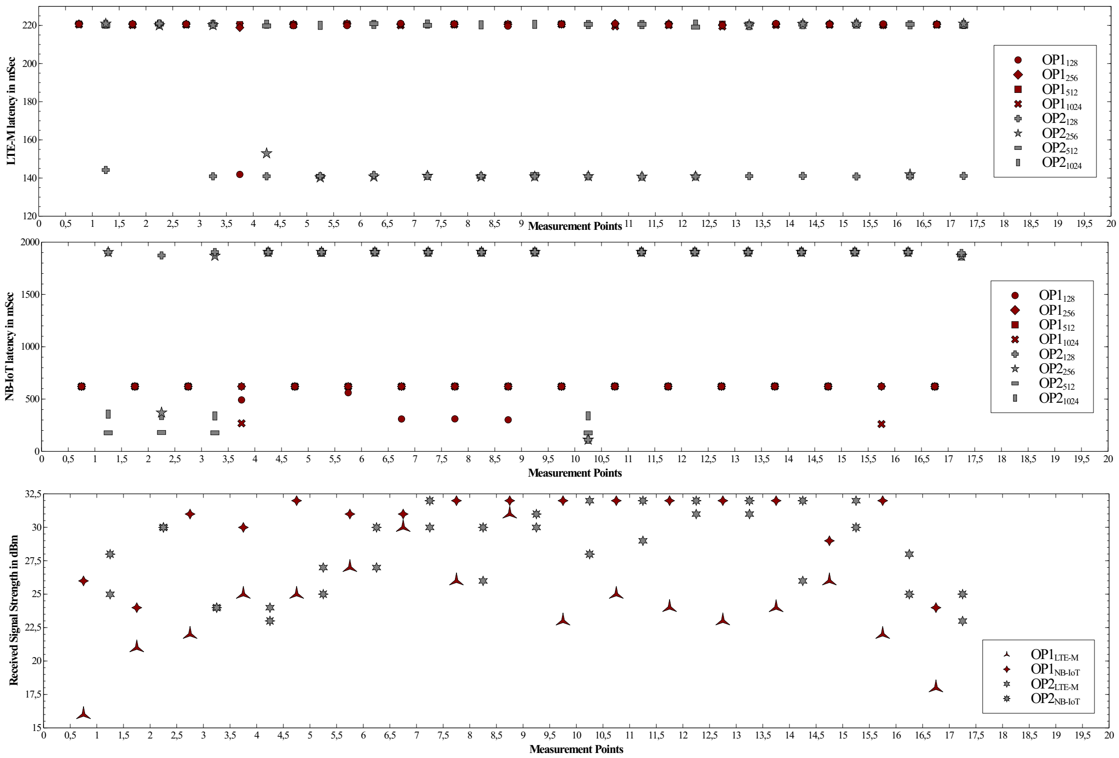


Figure 5.18: E2E Latency at all points outdoor environment

OP1 across all the points is 220 mSec. In the case of OP2, the E2E latency performance reflects the payload sizes of the test packets. For example, the tests involving packet sizes of 128 Bytes and 256 Bytes have shown lower E2E latency numbers than the 512 Bytes and 1024 Bytes. This is in contrast to what was observed in the 5.2.1, where the median E2E latency appeared to be independent of the payload size. This difference in the behaviour was not observed in the case of OP1 LTE-M network. The difference E2E latency performance in the case of OP2 LTE-M could be justified because of:

- Changes in the LTE-M RAN parameters of OP2 whch enabled more RAN resources for the LTE-M networks.
- The indoor tests described in section 5.2.1 were conducted during Christmas holiday period indicating differences in the availability of the RAN resources, that could justify the variation in the E2E latency.

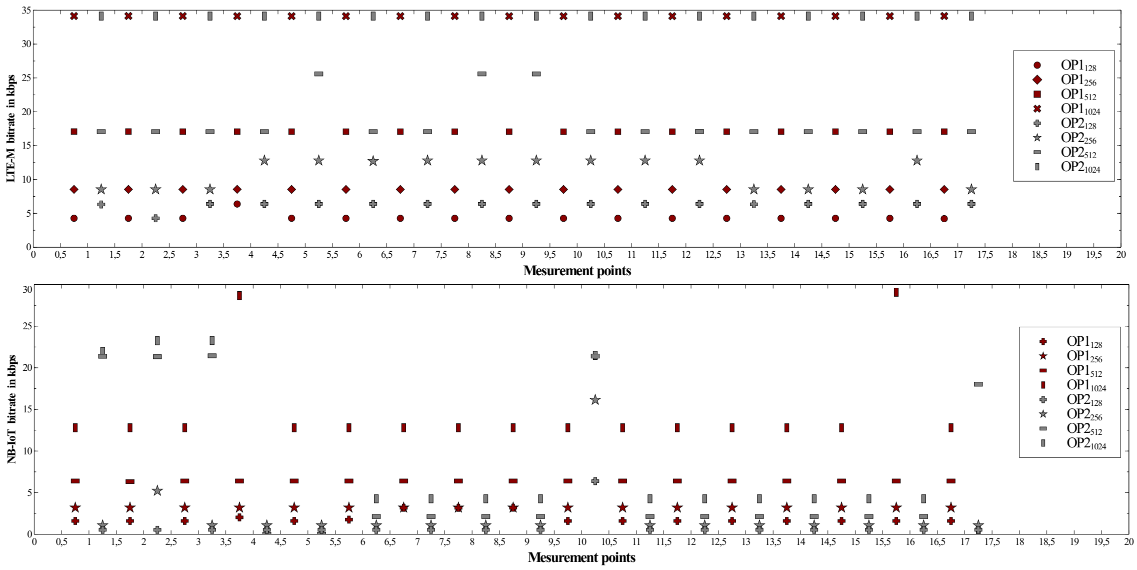


Figure 5.19: Bitrate at all points during outdoor environment

When it comes to the NB-IoT E2E latency performance, OP1 has a very consistent performance. In the case of OP1, the calculated median E2E latency across all the points is 612 mSec. A strange network behavior was observed while testing OP2 for E2E latency KPI using NB-IoT. For instance, at point 1, the initial latency recorded during the test is approximately 1.9 seconds (with payloads of 128 and 256 bytes). However, as the test progresses, the latency significantly decreases to lower values of 400 milliseconds (with a 512-byte payload) and 200 milliseconds (with a 1024-byte payload). This similar behavior is observed at points 2, 3, and 10, and repeated tests resulted in identical results. An initial possible hypothesis was that the device in the middle of testing was handed over to the LTE-M RAT as a part of the network feature. However, after checking the network metadata multiple times, the device consistently reported being connected to the NB-IoT RAT. The behavior was more suspicious, given that the PYCOM FIPY board was flashed with different C-IoT modem firmware versions for LTE-M and NB-IoT. This strange behavior was reported to the concerned MNO for further investigation. Several possible causes for such a behavior are listed below:

- This sudden behavior change was caused due to an error in the modem firmware.
- Possible availability of more radio resources while testing allowed low resource al-

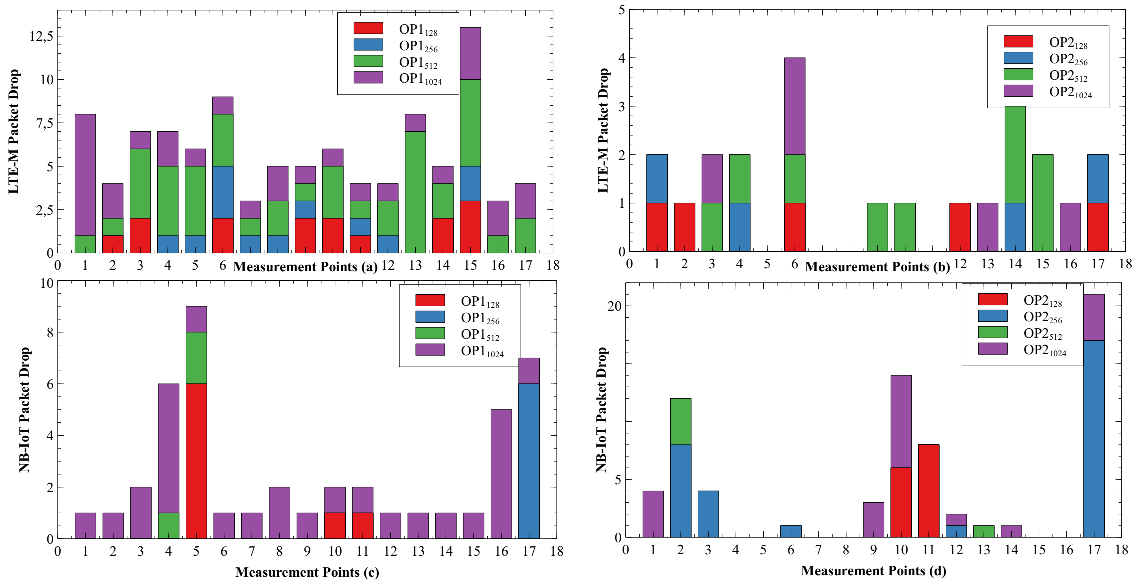


Figure 5.20: PD at all points outdoor environment

location delays.

- Error in the ratio parameter setting set by the MNO or possible bug in the NB-IoT eNB software.

Unfortunately, during these tests, it was impossible to access the MNO's network infrastructure and network traces. Therefore the concrete reason behind such behavior on NB-IoT RAT was never identified.

The graph reporting the received signal strength indicates a gradual increase in received power from points 1 to 13, followed by a subsequent decline. This observed behavior aligns with expectations, as the tests were intentionally designed to transition from a NLoS environment to a LoS environment, and then go back to the initial NLoS conditions.

Figure 5.19 shows the recorded bitrate performance observed during LTE-M and NB-IoT testing on OP1 and OP2 networks in outdoor environment. As can be observed from the plot for LTE-M, the calculated median bitrate goes up as the test packet payload size increases. This is an expected behavior since the increase in payload size increases the number of bits arriving at the application server. The highest observed median bitrate for OP1, considering all points, is 34 Kbps, while the lowest observed median bitrate is 4.7 Kbps. Similarly, in the case of OP2, the highest observed median bitrate is 33 Kbps, and the lowest observed median bitrate is 4.6 Kbps.

In the case of NB-IoT, the overall performance of both MNO is very similar and consistent. The highest median bitrate for OP1 is 31 Kbps, whereas, for OP2, it is 23 Kbps. The lowest median bitrate recorded for OP1 is 1.6 Kbps, and for OP2 is 0.8 Kbps. When comparing all the KPI values, OP2 has slightly worst performance across the board in comparison with OP1.

Figure 5.20 shows the recorded PD during testing LTE-M and NB-IoT using OP1 and OP2 in an outdoor environment. The bar plots are color-coded to give a PD at a given location divided into individual payload sizes. Figure a shows the PD recorded while testing LTE-M using OP1 in an outdoor environment. As can be seen from the graphs, the maximum

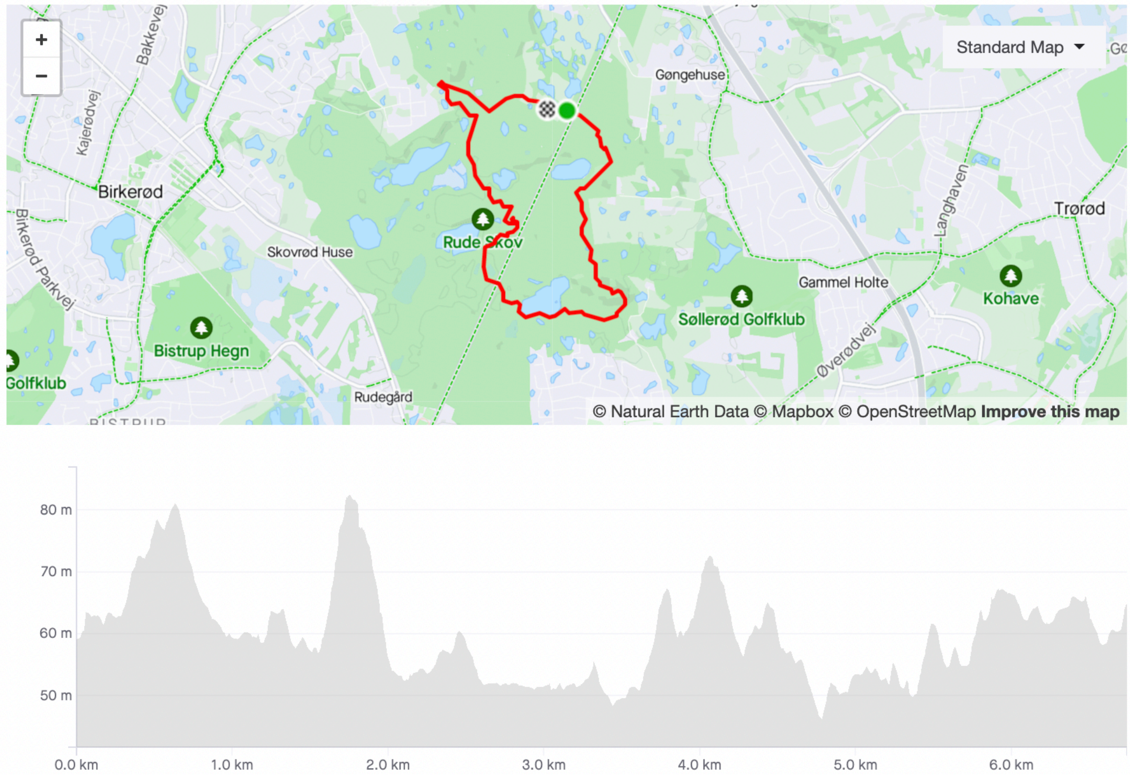


Figure 5.21: Forest measurement path and Elevation map [94]

PD occurred at point 15 with a loss of 13 packets, whereas the minimum PD of 3 packets was at points 7 and 16. In the case of OP2 LTE-M outdoor testing shown in figure b, the maximum PD of 4 packets was recorded at point 6, whereas the minimum PD of zero was recorded at points 5, 7, 8, and 11.

In the case of NB-IoT outdoor testing using OP1 plotted in figure c, the maximum PD was experienced at point 5 with 9 packets dropped, and the lowest PD was observed at multiple test points with 1 PD. Compared with OP2 shown in figure d, no PD is observed at points 4,5,7,8,15, and 16. OP2 has experienced the highest packet loss at point 17, with 20 packets dropped.

5.2.4 Remote Outdoor tests

The remote outdoor test environment was the final test environment where it was decided to perform the KPI testing. This test aimed to understand the behavior and the KPI performance of C-IoT technologies in remote outdoor locations where there is no direct LoS with any of the eNB and consist of a mixed terrine.

Figure 5.21 shows the path chosen to perform this test and the elevation of the terrine. The figure's red line indicates the path followed during the test. The total length of the path was approximately 6.74 km, and all the measurements were taken 500 meters apart. Therefore, there were 13 locations in which the tests were conducted. Test point 6 in the path was at the highest elevation of 83 meters, whereas point 11 had the lowest elevation of 46 meters.

Figure 5.22 shows the location of the C-IoT cell towers spread across the test path. All

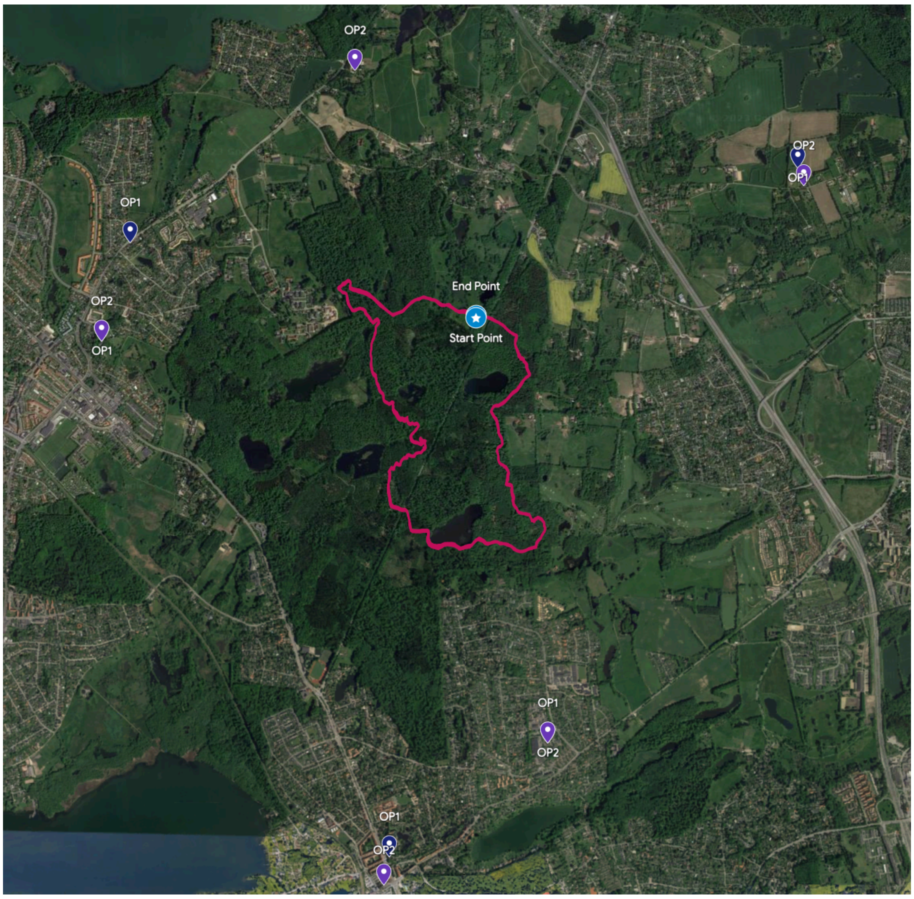


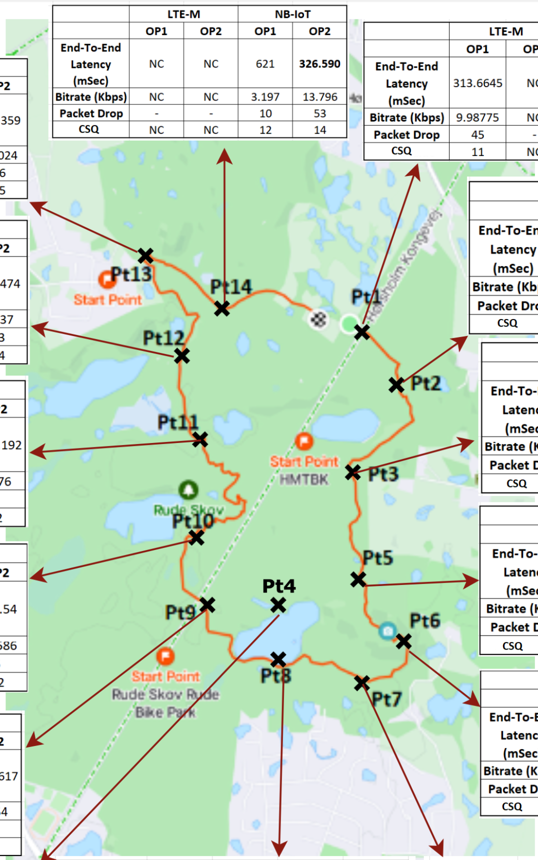
Figure 5.22: eNB Locations for OP1 and OP2

the eNB on the map are located between 0 km to 2.5km. As it can be observed from the locations of the eNBs, in the case of both the operators, there are no C-IoT eNBs in the radius of 1.3 km from the test points. Another fact is that, even though there are a few C-IoT cell towers on the outskirts of the forest, the direction of the eNB sectors is unknown.

Uneven test terrine, no LoS with the eNBs, and no eNBs within the 1.3km radius make this test very challenging, and it is reflected in the test results.

Figure 5.23 shows the different points along the path where these tests were conducted and the results obtained at each point. At each point, the table summarises the median E2E latency (in mSec), median bitrate (in Kbps), calculated PD, and CSQ. If there is no connectivity for a particular Operator or the C-IoT technology at any test point, it is denoted by No Coverage (NC) in the table. The tests are conducted, and KPI are extracted using the same test procedure described in Figure 5.3.

OP1 LTE-M: As seen from the different tables in Figure5.23, OP1 had coverage at 7 out of the 14 points on the test trail. Points 1, 2, and 3 had comparable performance, and point 3 had the best E2E latency and bitrate of the three. Point 2 observed the lowest drop of 5 packets and had the highest RSRP value. The higher E2E latency and bitrate performance at points 2 and 3 could be because the elevation in the terrine allowed devices to be above the tree levels and somewhat LoS with the eNB to the test of the test path. At Points 5 and 6, there was no coverage for the OP1 LTE-M signal, and the tests could not be conducted. This could have been because point 5 is somewhat is a valley and



	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	NC	704.241	301.324	401.359
Bitrate (Kbps)	NC	3.869	9.024	10.024
Packet Drop	-	2	16	46
CSQ	NC	17	13	15

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	NC	NC	621	**326.590**
Bitrate (Kbps)	NC	NC	3.197	13.796
Packet Drop	-	-	10	53
CSQ	NC	NC	12	14

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	313.6645	NC	274.3275	1855.294
Bitrate (Kbps)	9.98775	NC	12.283	1.6175
Packet Drop	45	-	50	9
CSQ	11	NC	13	12

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	221.23325	218.280	230.1125	1897.548
Bitrate (Kbps)	12.7805	12.803	15.746	0.8
Packet Drop	5	0	5	0
CSQ	13	19	18	14

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	NC	NC	661.393	447.474
Bitrate (Kbps)	NC	NC	4.733	6.537
Packet Drop	-	-	22	13
CSQ	NC	NC	15	14

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	220.66	NC	NC	343.12
Bitrate (Kbps)	18.795	NC	NC	9,369
Packet Drop	47	-	-	4
CSQ	10	NC	NC	11

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	NC	NC	633.486	1138.192
Bitrate (Kbps)	NC	NC	4.656	9.776
Packet Drop	-	-	0	6
CSQ	NC	NC	11	12

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	NC	NC	467.951	1906.149
Bitrate (Kbps)	NC	NC	8.685	1.6
Packet Drop	-	-	5	0
CSQ	NC	NC	0	15

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	NC	NC	802.486	333.54
Bitrate (Kbps)	NC	NC	1.836	10.586
Packet Drop	-	-	2	5
CSQ	NC	NC	10	12

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	NC	NC	620.726	275.968
Bitrate (Kbps)	NC	NC	4.8	15.736
Packet Drop	-	-	0	4
CSQ	NC	NC	20	15

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	201.632	NC	620.563	1195.617
Bitrate (Kbps)	16.408	NC	4.799	2.654
Packet Drop	25	-	0	76
CSQ	11	NC	14	12

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	231.94	NC	620.834	1906.95
Bitrate (Kbps)	12.693	NC	4.799	3.1955
Packet Drop	18	-	0	0
CSQ	13	NC	15	17

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	221.086	NC	620.709	1334.431
Bitrate (Kbps)	12.793	NC	4.801	0.584
Packet Drop	2	-	0	0
CSQ	11	NC	14	11

	LTE-M		NB-IoT	
	OP1	OP2	OP1	OP2
End-To-End Latency (mSec)	220.596	NC	620.726	260.386
Bitrate (Kbps)	12.798	NC	4.799	18.924
Packet Drop	5	-	1	8
CSQ	13	NC	11	15

Figure 5.23: Forest C-IoT test results in Remote outdoor environment

point 6 had the highest elevation in the test path, and there are no eNBs from OP1 directly facing toward those points. Points 7, 8, 9, and 4 have a consistent coverage of LTE-M from OP1. These are also the points that are closer to the eNB from the OP1. Point 9 has the lowest E2E latency, 201.632 mSec, and the highest bitrate of 16.04 Kbps and PD of 25 packets. Point 9, being closer to the road, might be one of the contributing factors. Points 10, 11, 12, 13, and 14 had no OP1 coverage to perform the KPI testing. This could be because the signal from the closest OP1 eNB might be too noisy. There might be no sectors of the eNBs facing toward the test path resulting in an outage scenario.

OP1 NB-IoT: As it can be observed from all the results, OP1 NB-IoT coverage was much stronger than that of LTE-M. Only at point 3 did the OP1 NB-IoT test fail due to insufficient coverage. The test was repeated multiple times at point 3, resulting in no coverage. In the case of OP1 NB-IoT, points 1, 2, 5, and 13, the recorded E2E latency was less than 500 mSec, point 2 being the lowest at 230.1125 mSec. Point 2 was also the point at which the highest bitrate was recorded, with 15.746 Kbps. No PD was recorded at points 4, 6, 8, 9, and 11. The highest PD was recorded at point 1, with 50 dropped packets. OP1 NB-IoT shows the characteristics of NB-IoT standards and coverage improvements promised being translated into realworld performance.

OP2 LTE-M: The performance of LTE-M from OP2 was the worst overall. Out of 14 points, it was only possible to conduct the KPI testing at 2 points. During the test setup on many locations, the device was getting connected to the network but was getting de-registered right after sending a few data packets. This behavior was very strange, given that several eNB sites were deployed around the test path. Points 2 and 13 were the only points where the KPI tests were successful, and point 2 had the best performance KPI of the two. At the point to the recorded E2E latency was 218.280 mSec, and the bitrate was 15.746 Kbps, but it had a higher packet loss of 5.

OP2 NB-IoT: On the contrary to the OP2 LTE-M performance, the OP2 NB-IoT performance was better. It was possible to conduct the KPI testing on all 14 points using the OP2 NB-IoT. There was a minor change in the test configuration when it came to NB-IoT testing, and the device was sending approximately 1/3 of the packets as that of the OP1 NB-IoT tests. This change was made because it took a long time to get the initial network attached at the beginning of the test and longer E2E latency. Points 1, 2, 3, and 5 have very similar performance, except, at point 3, the E2E latency of 343.12 mSec is much lower than the rest, and the bitrate is quite high, 9.369 Kbps. The RSRP values are above 10 dBm. Points 6 & 7 have very similar in terms of KPI performance, point 7 being better by a small margin. At point 7, the recorded E2E latency was the lowest of all the points, with 260.486 Kbps and the highest bitrate of 18.924 Kbps. Points 4, 8, and 9 all have E2E latency values higher than 1 second. Point 4 has the highest recorded E2E latency value of 1.906 seconds and the highest bitrate of 3.1955 Kbps. Point 9 has the highest packet loss of all the test points, with 76 overall dropped packets. Points 10, 11, 12, and 13 had much better overall OP2 NB-IoT performance except for point 11, where the E2E latency was an average of 3 times higher than the rest with 1.138 mSec.

Overall performance of both the MNO on LTE-M and NB-IoT is described in table 5.3.

5.3 C-IoT performance over time

This section highlights the results from testing the C-IoT over time. The idea behind these tests was to determine any changes in the calculated performance of the NB-IoT and LTE-

	E2E latency (mSec)		Bitrate (Kbps)		PD	Total Packets sent	Drop Rate
	Min	Max	Min	Max			
OP1 LTE-M	201.632 (Point 9)	313.6645 (Point 1)	9.98775 (Point 1)	18.795 (Point 3)	147	26715	0.55%
OP1 NB-IoT	239.1125 (Point 2)	802.486 (Point 10)	1.836 (Point 10)	15.746 (Point 2)	111	24359	0.45%
OP2 LTE-M	218.28 (Point 2)	704.241 (Point 13)	3.869 (Point 13)	12.803 (Point 2)	2	4846	0.04%
OP2 NB-IoT	260.486 (Point 7)	1906.95 (Point 4)	0.584 (Point 8)	18.924 (Point 7)	218	17950	1.21%

Table 5.3: Summery of Forest C-IoT test results in Remote outdoor environment

M when the measurements are performed three months apart. The hypothesis was to observe possible changes in the C-IoT performance caused by network software updates, radio replanning, changes in network parameters, etc. The tests were performed with a smaller sample size than those performed in section 5.2. The test setup was the same, with no changes from the UE and remote server side. The test locations were also kept the same to minimize any measurement error. The only difference between the two tests was the timeframe for the tests. All the tests conducted in this section are approximately 90 days after the initial network testing.

				Point -1	Point -5	Point -9	Point -13	Point -17
OP-1	**LTE-M**	**E2E latency**	**Old**	314.351	330.92225	220.0355	171.2095	258.836
			New	NC	NC	NC	NC	NC
		bitrate	**Old**	50.6512	8.50145	12.798	21.615	10.73675
			New	NC	NC	NC	NC	NC
	NB-IoT	**E2E latency**	**Old**	169.5799	**404.0555**	620.6152	669.036	**620.603**
			New	NC	619.3665	**619.005**	**619**	1383.168
		bitrate	**Old**	16.564	**4.79875**	**4.8**	4.1317	**4.8**
			New	NC	2.3987	4.7987	**4.8**	0.74
OP-2	**LTE-M**	**E2E latency**	**Old**	257.36	NC	168.015	203.329	NC
			New	NC	NC	**140.88**	**88.975**	NC
		bitrate	**Old**	7.543	NC	15.2632	16.408	NC
			New	NC	NC	**19.2**	**26.846**	NC
	NB-IoT	**E2E latency**	**Old**	**265.965**	1855.188	1048.177	1904.636	1456.053
			New	2896.69	NC	**286.2295**	**271.7905**	NC
		bitrate	**Old**	**10.4889**	0.775	7.5747	1.5995	6.9537
			New	0.318	NC	**17.8995**	**17.2792**	NC

Table 5.4: Performance comparison deep indoor scenario (measurements are three months apart)

Table 5.4 shows the performance comparison between the tests conducted in a deep indoor scenario. The tests were conducted in the underground tunneling system at DTU at locations that were 400 meters apart. The E2E latency is measured in mSec whereas the bitrate is measured in Kbps.

As it can be observed from the table, there was no coverage detected for LTE-M from OP1. This was completely contracting behavior to the initial OP1 LTE-M performance. Similarly, in the case of NB-IoT from OP1, the performance is degraded since the earlier measurements in 5.2. In the case of OP2, the change in performance is less drastic.

				Point -1	Point -5	Point -9	Point -13	Point -17
OP-1	LTE-M	PD	Old	3045	8	12	145	128
			New	NC	NC	NC	NC	NC
	NB-IoT	PD	Old	3	5	6	62	24
			New	NC	0	0	0	34
OP-2	LTE-M	PD	Old	7518	NC	1	55	NC
			New	NC	NC	0	4	NC
	NB-IoT	PD	Old	144	1285	114	3728	40
			New	52	NC	16	5	NC

Table 5.5: PD performance comparison deep indoor scenario (measurements are three months apart)

				Point -1	Point -5	Point -9	Point -13	Point -17
OP-1	LTE-M	E2E latency	Old	220.7357	220.415	220.4967	220.12	220.5112
			New	NC	NC	NC	NC	NC
		bitrate	Old	12.8085	12.8	12.797	12.798	12.801
			New	NC	NC	NC	NC	NC
	NB-IoT	E2E latency	Old	620.621	620.3022	620.5407	621.1872	620.3275
			New	NC	619.1795	619.05675	619.20825	619.2082
		bitrate	Old	4.7995	4.8	4.7995	4.8	4.799
			New	NC	3.1985	4.8	4.8	3.2
OP-2	LTE-M	E2E latency	Old	220.291	140.9185	141.365	220.1127	220.0667
			New	NC	181.0095	179.747	**180.48275**	180.996
		bitrate	Old	12.8	19.1947	19.193	12.7995	12.8005
			New	NC	12.796	14.93825	**14.933**	12.799
	NB-IoT	E2E latency	Old	1130.0877	1904.115	1904.644	1904.95	1882.2245
			New	NC	1904.5775	1904.9977	1904.896	1904.923
		bitrate	Old	11.2265	1.6028	1.5995	1.5995	1.068
			New	NC	1.066	1.5995	1.5995	1.066

Table 5.6: Performance comparison outdoor (measurements are three months apart)

At point 1 for LTE-M and at points 5 & 7 for NB-IoT, there is no longer C-IoT coverage detected. In the case of LTE-M coverage from OP2, the rest of the point has either stayed the same or shown slight improvement. In the case of NB-IoT from OP2, the network has seen performance increase at points 9 & 10 and performance degradation at point 1.

Table 5.5 shows the dropped packets recorded while performing the tests. Similar to the E2E latency and bitrate tests, OP1 LTE-M had no PD data available. In the case of NB-IoT for OP1, the PD performance slightly improved with fewer dropped packets. Similarly, in the case of OP2, the number of dropped packets was reduced for LTE-M and NB-IoT.

Table 5.6 shows the performance comparison of C-IoT networks in an outdoor scenario. The tests were conducted at the exact location as that of in section 5.2, and the points were 400 meters apart. The E2E latency is measured in mSec whereas the bitrate is measured in Kbps.

Similar to the deep indoor testing, there was no network coverage offered by OP1 for its LTE-M network. This was the opposite network behavior than that of the tests conducted earlier. In the case of the NB-IoT network from OP1, except for point 1, where there was no NB-IoT coverage, the rest of the network performance was very similar with some minor variations.

				Point -1	Point -5	Point -9	Point -13	Point -17
OP-1	LTE-M	PD	Old	8	6	5	8	4
			New	NC	NC	NC	NC	NC
	NB-IoT	PD	Old	1	9	1	1	7
			New	NC	0	1	0	0
OP-2	LTE-M	PD	Old	2	0	1	1	2
			New	NC	0	0	0	0
	NB-IoT	PD	Old	4	0	3	1	21
			New	NC	0	0	1	0

Table 5.7: PD performance comparison outdoor (measurements are three months apart)

Similarly, for OP2, no C-IoT coverage was observed at point 1. In the case of LTE-M at points 5 & 9, the performance was slightly degraded compared to previous measurements. Whereas at points 13 & 17, it was improved. Similarly, in the case of NB-IoT, the performance remained identical except for improvements at point 5.

Table 5.7 shows the PD performance of C-IoT networks while testing in an outdoor environment. Similar to the E2E latency and bitrate tests, there was no data available due to lack of coverage in the case of OP1 LTE-M network. For NB-IoT offered by OP1, the PD performance slightly improved in comparison with the initial testing. PD performance also improved for C-IoT networks offered by OP2 in an outdoor scenario.

				Point -1	Point -4	Point -7	Point -10	Point -13
OP-1	LTE-M	E2E latency	Old	313.6645	231.94	220.596	NC	NC
			New	185.8	217.1855	221.10625	313.88775	NC
		bitrate	Old	9.987	12.693	12.798	NC	NC
			New	13.18975	12.832	12.778	8.9605	NC
	NB-IoT	E2E latency	Old	274.32	620.834	620.726	802.486	301.324
			New	630.0037	500.9905	619.00875	648.627	737.0992
		bitrate	Old	12.283	4.799	4.799	1.836	9.024
			New	4.734	7.05625	4.79975	3.2845	3.6805
OP-2	LTE-M	E2E latency	Old	NC	NC	NC	NC	704.241
			New	NC	NC	NC	NC	NC
		bitrate	Old	NC	NC	NC	NC	3.869
			New	NC	NC	NC	NC	NC
	NB-IoT	E2E latency	Old	1855.294	1906.95	260.386	333.54	401.359
			New	1766.1007	554.7235	1875.426	591.25575	NC
		bitrate	Old	1.6175	3.1955	18.924	10.586	10.024
			New	5.09925	4.62525	2.34025	6.122	NC

Table 5.8: Performance comparison remote outdoor (measurements are three months apart)

Table 5.8 compares performance C-IoT measurements taken three months apart in remote outdoor conditions. The measurement points are aligned with the measurement points from the initial tests in section 5.2. The E2E latency is measured in mSec whereas the bitrate is measured in Kbps.

In the case of OP1 LTE-M network, the performance has improved compared to the initial measurements, especially at point 10, where there is now LTE-M coverage. Similarly, in the case of NB-IoT coverage from OP1, network improvements were recorded at points 4 & 10, whereas performance degradation was recorded at points 1 & 13.

				Point -1	Point -4	Point -7	Point -10	Point -13
OP-1	**LTE-M**	**PD**	**Old**	45	18	5	NC	NC
			New	13	0	0	8	NC
	NB-IoT	**PD**	**Old**	50	0	1	2	16
			New	0	51	0	8	0
OP-2	**LTE-M**	**PD**	**Old**	NC	NC	NC	NC	53
			New	NC	NC	NC	NC	NC
	NB-IoT	**PD**	**Old**	9	76	8	5	45
			New	23	69	1	12	NC

Table 5.9: PD performance comparison remote outdoor (measurements are three months apart)

			Point -1	Point -5	Point -9	Point -13	Point -17
OP-1 (LTE-M)	E2E latency	Old	220.7357	220.415	220.4967	220.12	220.5112
		New	220.633	221.4612	221.551	221.0725	221.4717
	bitrate	Old	12.8085	12.8	12.797	12.798	12.801
		New	12.78025	12.7785	12.78	12.7752	12.79425

Table 5.10: Performance comparison outdoor for OP1 LTE-M network

The LTE-M network offered by OP2 saw a performance degradation. The LTE-M performance of OP2 during the initial testing was already poor, and three months later, it got even worse. In the case of the NB-IoT network from OP2, the network performance at points 1 & 4 has improved, whereas at points 7, 10, and 13 has degraded in comparison with initial tests.

Table 5.9 compares dropped packets in remote outdoor conditions. As can be observed from the table, the performance of LTE-M from OP1 improved slightly with fewer PDs. The NB-IoT network from OP1 improved a little except for the increased PD at point 4. Similar to the E2E latency and bitrate tests, the performance of LTE-M from OP2 has gotten worse with no coverage on the test locations. In the case of NB-IoT, the performance degraded slightly with higher PD observed at points 1 & 10 and with no connection at point 13.

When looking at the tests in deep indoor (table 5.4) and outdoor scenario (table 5.6) OP1 LTE-M tests could not be concluded. While testing, it was observed that the device could not attach to the OP1 network. This was an abnormal behavior recorded while testing and an implausible scenario. Therefore, it was decided to perform the tests again using OP1 LTE-M network in an outdoor environment; the tests were conducted three days after the initial testing. Table 5.10 shows the results from the tests conducted using LTE-M network of OP1. As seen from the tests, it was possible to perform the LTE-M tests at all the test points without any issues. This indicated some issue with the OP1 LTE-M network.

In order to further investigate this issue, it was decided to perform all the tests once more at point 1, given that all the tests failed at this point. After looking at the test data, it was discovered that all the tests were completed successfully at point 1 from both the MNO using C-IoT.

A few possible reasons could have been responsible for this network behavior:

- The network blocked the test devices for a limited time, leading to no network attach.
- There was C-IoT network outage in the eNB clusters covering DTU.

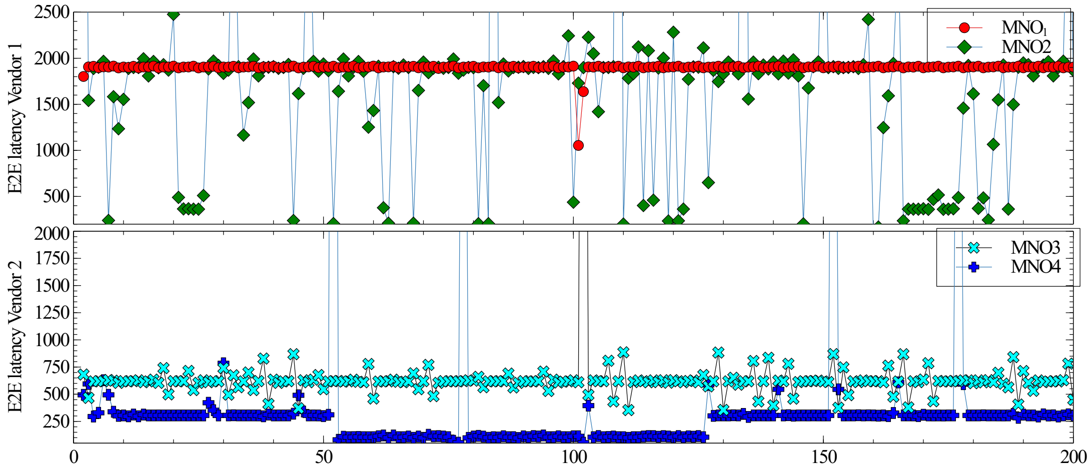


Figure 5.24: NB-IoT E2E latency performance using different RAN vendors

- There was a power outage that caused C-IoT networks from both the MNOs to become completely unavailable from the eNB cite

This recorded issue needed further investigation, but unfortunately, a concrete reason for this behavior was not identified due to lack of access to the MNO infrastructure and device traces.

Another test performed during this timeframe was to identify KPI performance of C-IoT networks deployed by different MNO using different RAN vendors. Figure 5.24 shows the observed difference between four different Nordic MNO and E2E latency performance of their NB-IoT based on the RAN vendor that is used to deploy the radio network. The figures displays unprocessed E2E latency values measured over 200 UDP packets of 128 B each. The RAN vendor 2 E2E latency performance is superior than that of the RAN vendor 1. The similar difference in the performance are measured in the case of bitrate and PD experiments.

5.4 C-IoT Network Roaming Tests

Expain why roaming and how is it relevant

In order to evaluate the KPI performance of C-IoT, MNO was selected, which has nation-wide C-IoT deployment in Scandinavia. The test was conducted to evaluate the performance of the IoT test device in the home network and then compare the performance in two different roaming networks. The tests were conducted in Denmark (Home Network), Sweden (Operator Roaming Network), and Norway (Operator Roaming Network). The test setup and procedure were the same as described in section 5.1 C-IoT test setup and Network KPI. In all the tests conducted for this experiment following steps were followed:

- The tests were performed in indoor environments, e.g., University offices, coffee shops, etc.
- The tests were always conducted in the good coverage area with RSRP > -85 dBm.
- To avoid any errors caused by network parameters, all the MNO-specific features were turned off. e.g., power-saving features such as PSM, eDRX, cDRX, etc., were

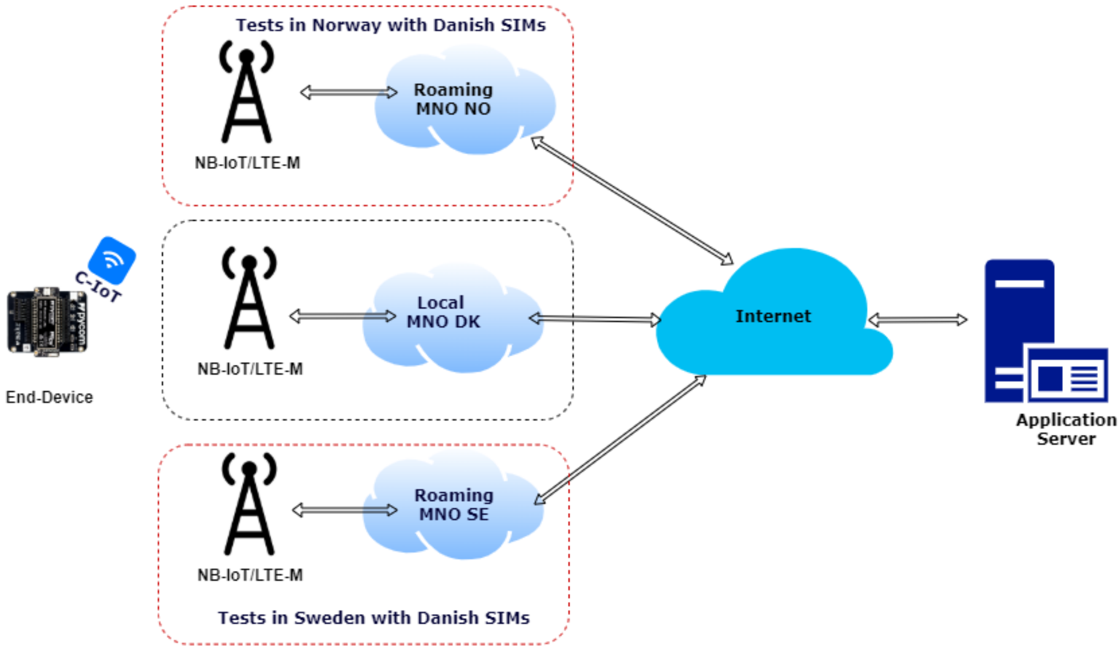


Figure 5.25: C-IoT Roaming Test Network Architecture [95]

turned off before the test started.

- All the tests were conducted using only 2 PYCOM FIPY devices, one for LTE-M and one for NB-IoT.
- The network parameters, such as APN name, were locked to Danish MNO settings while testing the device in Norway and Sweden.
- In all the tests, the device was programmed to select the operator automatically.

Figure 5.25 shows the C-IoT KPI test architecture used in the roaming environment. As can be observed from the figure, the test end device and the application server were kept constant across all the tests. The application server was hosted in Frankfurt, Germany. Therefore the application server was approximately 825 km from the Danish test site, 869 km from the Swedish test site, and 1800 km from the Norwegian test site (Distances are calculated using Google Maps). Therefore, the only variable in all the tests was each country's individual C-IoT MNO network.

The first tests were conducted in Denmark at the Technical University of Denmark. The tests were conducted using a Danish SIM card. This was the regular test scenario where a local MNO SIM card connected to its home network. The test was conducted by sending 1000 packets of varying packet sizes 128 Bytes, 256 Bytes, 512 Bytes, and 1024 Bytes. The packets were sent using UDP as a transport protocol.

The same device was then tested in Sweden under similar conditions. The device was tested at Lund University premises with a Danish SIM card. This meant that the device was sending data using the C-IoT network in Sweden but with a Danish SIM, which was the first roaming operator. The tests were the same as those conducted in Denmark, with 1000 UDP packets with varying payloads sent from each of the C-IoT technologies.

In order to perform roaming KPI testing in Norway, the same setup was used as that of Denmark and Sweden. The tests in Norway were performed in similar physical conditions at the Norwegian University of Science and Technology (NTNU) premises. The test in-

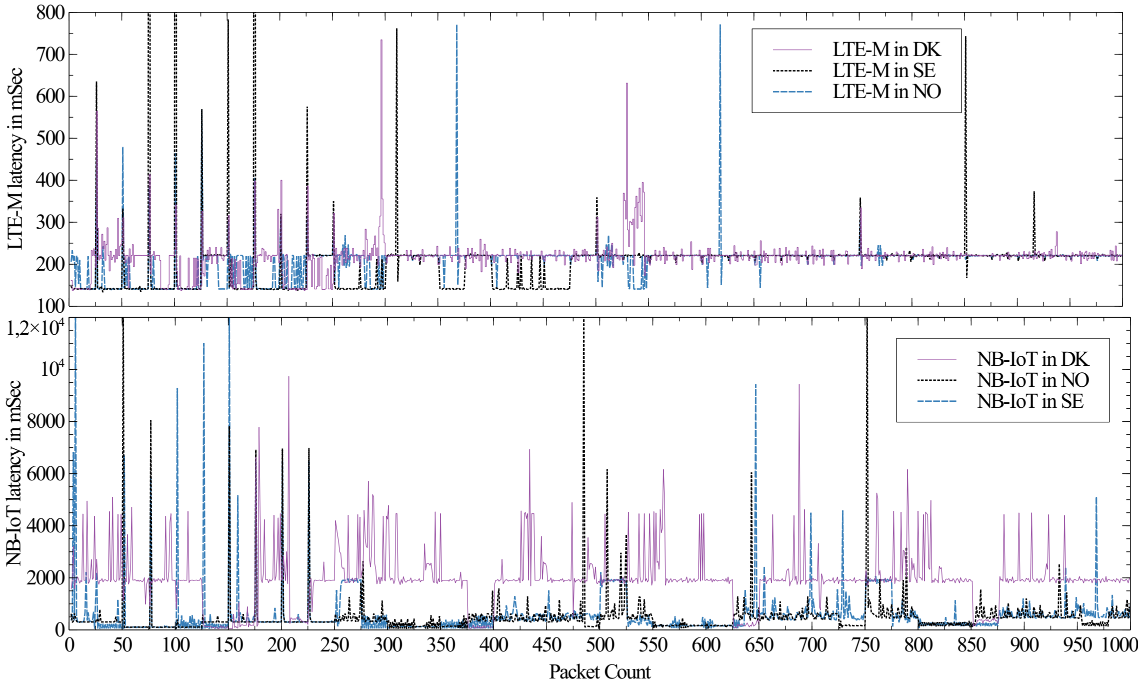


Figure 5.26: C-IoT E2E Latency Tests in Roaming Environment [95]

cluded sending 1000 UDP packets with varying packet sizes over Norwegian C-IoT MNO using a Danish SIM card.

Figure 5.26 shows the overall observed E2E Latency during the C-IoT KPI testing in Denmark, Sweden, and Norway. The graph's X-axis shows the packet count, whereas the Y-axis shows the calculated E2E Latency.

In the case of LTE-M, the E2E latency performance across all the countries is very similar. At the beginning of the test for the packets with a payload size of 128 Bytes, the performance is very unstable in all the countries, but for the higher payload sizes, the E2E latency is stable with some exceptions where the latency suddenly increases or decreases than the median value. In the case of NB-IoT, the performance of the network in all the countries is not very stable, and unlike LTE-M, there are no visible patterns observed during the testing. There are several spikes observed in E2E latency performance in all three countries. Another observation is that the E2E latency performance recorded in the Danish network is much higher than that of Sweden and Norway. In Sweden and Norway, the median E2E latency performance is much closer to the median LTE-M E2E latency performance.

Figure 5.27 shows the calculated bitrate performance from C-IoT KPI testing in Denmark, Sweden, and Norway. The graph's X-axis shows the packet count, whereas the Y-axis focuses on bitrate values in Kbps. While testing LTE-M, the bitrate performance in all the countries is very similar, with some exceptions where the bitrate fluctuates between packets. As can be observed from the graph, the LTE-M birate increases in steps after every 250 packets. This is because the increase in the packet size from 128 Bytes to 1024 Bytes increases the number of bits arriving at the application server per minute. The overall LTE-M performance in Denmark, Sweden, and Norway is very similar and stable. In the case of NB-IoT KPI testing, similar to E2E latency testing, the calculated bitrate is very unstable, and no performance pattern can be extracted from the results. The noticeable

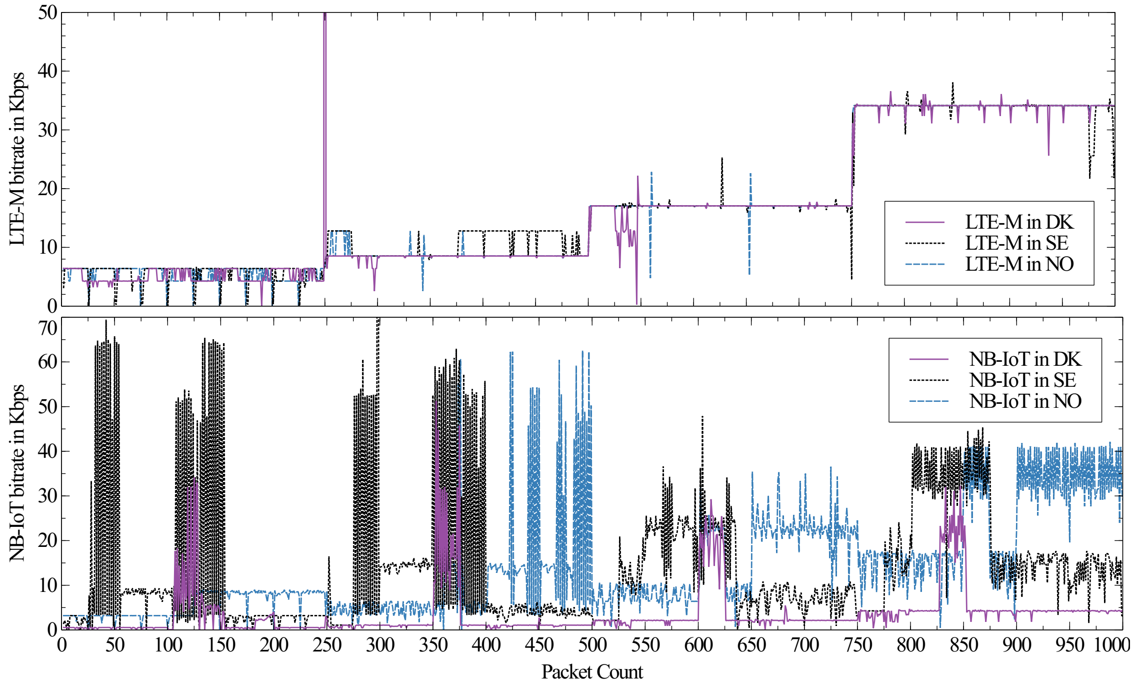


Figure 5.27: C-IoT bitrate in Roaming Environment [95]

Technology	E2E Latency (mSec)			Bitrate (Kbps)			Number of Packets Dropped		
	Denmark (Home Network)	Sweden (Roaming MNO)	Norway (Roaming MNO)	Denmark (Home Network)	Sweden (Roaming MNO)	Norway (Roaming MNO)	Denmark (Home Network)	Sweden (Roaming MNO)	Norway (Roaming MNO)
LTE-M	220.362	219.795	220.623	8.5665	12.8115	12.8055	36	16	40
NB-IoT	1905.004	373.571	314.77	2.102	8.7515	8.9805	180	64	31

Table 5.11: C-IoT Summery of Netowrk KPI in Roaming Environment [95]

difference was observed in the bitrate performance in the home network, i.e., Denmark was much lower than that of the roaming networks, i.e., Sweden and Norway. In Sweden, for the first 400 packets, the bitrate recorded bursts of packets with higher performance than the LTE-M bitrate at the same location. Similar behavior was also present in results from Norway; after 425 packets, higher bursts of bitrate were recorded where the bitrate values were very close to and even higher than the LTE-M bitrate at the same location.

Table 5.11 highlights all the median KPI values obtained from the different tests conducted in Denmark, Sweden, and Norway. As one can see from the table, the E2E latency performance of LTE-M in the home network (Denmark) and roaming networks (Sweden and Norway) is almost equal. Unfortunately, this is not the case when it comes to NB-IoT E2E latency performance. The E2E latency is worst in Denmark, which is the home network, compared to Sweden and Norway. Norway has the best NB-IoT E2E latency performance of all three countries.

In the case of bitrate performance, the device performs the worst in the home network for both LTE-M and NB-IoT. The bitrate performance in the roaming environment is very similar, i.e., the LTE-M and NB-IoT calculated median bitrate values in Sweden and Norway are almost equal for the respective technologies.

When it comes to the PD, comparing the LTE-M KPI in all three countries, Norway has recorded the highest PD of 40 packets, and Denmark is a close second with 36 packets. When considering the NB-IoT PD, the results in the home network are more than 2.9 times the worst performer in the roaming network.

After comparing all the KPI results from the different tests conducted in Denmark, Sweden, and Norway, it was observed that the LTE-M performance offered by this MNO across its Scandinavian footprint is comparable. However, this is different when it comes to the NB-IoT performance from the same MNO in all three countries.

After analyzing the results further, we could develop the following hypotheses.

- The NB-IoT network configuration implemented, especially the radio parameters in the different countries, are different.
- The NB-IoT deployment method, i.e., guard band, in-band, etc., could be different in different countries
- The Network resources (both the radio and the rest of the network) reserved for C-IoT technologies might be different in each country.

None of the above hypotheses could be further investigated due to the confidentiality around network architecture, deployment strategies, etc. Nonetheless, these exercises gave a deeper insight into how C-IoT roaming works and the key elements to consider while developing a battery-powered, constrained IoT device.

5.5 Network capacity tests for continuous data transmission

One of the features highlighted in the 3GPP standards for C-IoT is the ability to support a massive number of end-devices per cell cite sector [96, 97]. According to the 3GPP, C-IoT device density per sector is calculated to be 52547 devices. This calculation assumes the number of devices per household to be 40 and data throughput of 50 bytes every two hours per device [97]. A simulation study by Ericsson shows that the devices supported by per NB-IoT cell site are up to 200,000 [36]. The capacity of deployed devices per cell increases to 72000 in an in-band NB-IoT deployment study performed by Nokia [98]. In a study by Mads *et al.* [99], the cell sector capacity for two applications was evaluated. During the day, depending upon the type of application and the number of transmissions, the total number of devices supported by a cell sector under challenging radio conditions by LTE-M ranges from 80K to 1 Million. In contrast, for NB-IoT, it ranges from 5K to 25K devices. Tin *et al.* [100] proposes a new Periodic Uplink Scheduling Algorithm (PPUSA) that considers the PSM features offered by the technology and can increase the number of connected devices to 600,000 in case of NB-IoT.

In all the simulations studies conducted to calculate the cell sector capacity of C-IoT device, it was assumed that the device would only transmit a few times during the day, and no studies were available defining the continuous data transmission capacity of the cell sector. Since the application designed during this PhD relies on continuous ECG and HR monitoring of CVD patients, it was decided to perform an experimental evaluation of cell sector capacity of NB-IoT and LTE-M networks.

In order to carry out the experimental evaluation of the cell capacity, a total of 8 tests were conducted for LTE-M and NB-IoT, respectively. In each test, the number of connected devices increased by one; e.g., test 1 was performed using only one IoT device, whereas test 2 was performed using two connected devices. The overall test setup is as follows:

- The tests were performed using the PyCOM Gpy devices. The device and the cellular module were updated to the latest available firmware.
- The devices were kept in an indoor environment with an NLoS with the nearest eNB. The device location was the same as the indoor tests described in Section 5.2.1.
- The devices were programmed to send 500 packets in each test with varying payload sizes (128 Bytes to 1024 Byes).
- The experiments were conducted using the same setup used in all other experiments and is described in 5.3.
- In all the experiments, latency, bitrate, PD, test duration, and number of re-connections were measured.

The device re-connection flag was used in this experiment due to the device's very fluctuating performance while testing the NB-IoT network. The reconnect flag was set if one of the following errors occurred and the modem had to be reset and reconnected again:

1. Continuous drop of more than ten packets.
2. modem is stuck and caused a "modem suspend error".
3. Complete network disconnection

In the case of NB-IoT tests, sometimes the device abruptly disconnects from the network. The only way to reconnect was to perform a power cycle reset, where the device was disconnected and reconnected again. Such power cycle attempts were also recorded each time the device was power cycled. This issue was observed only during the NB-IoT testing and not during the LTE-M tests.

Figure 5.28 shows the calculated E2E latency during all the network capacity tests performed. The X-axis represents the number of connected devices, whereas the Y-axis represents the calculated median latency in different tests.

In the case of test 1, the lowest latency calculated was 295 mSec, whereas the highest latency calculated was 545 mSec.

For test 2, the lowest latency of 296 mSec was calculated by device d2, whereas the highest latency of 545 mSec was calculated by d1.

In test 3, the E2E is much lower for devices 2 and 3 than for device 1. In the case of device 1, the highest E2E is 544 mSec, similar to that of test 2, whereas, for devices 2 and 3, the E2E is 220 mSec.

Similarly, in the case of test 4, d1 and d3 have calculated higher E2E values compared to devices d2 and d4. The highest E2E latency value was 546 mSec by d3, whereas the lowest E2E latency value of 220 mSec by device d2.

In the case of test 5, d1 and d3 had higher E2E latency values than d2, d4, and d5. The highest E2E value of 547 mSec was calculated by d1, whereas the lowest E2E value of 220 mSec was recorded by d2.

During test 6, except d1, all the other devices had a lower E2E value. The highest E2E value of 544 mSec was calculated by d1, whereas the lowest E2E latency value of 220 mSec was calculated by d2 and d3.

In the case of test 7, devices d1, d2, d3, and d6 had higher E2E latency performance than that of devices d4, d5, and d7. The highest E2E latency value of 564 mSec was recorded

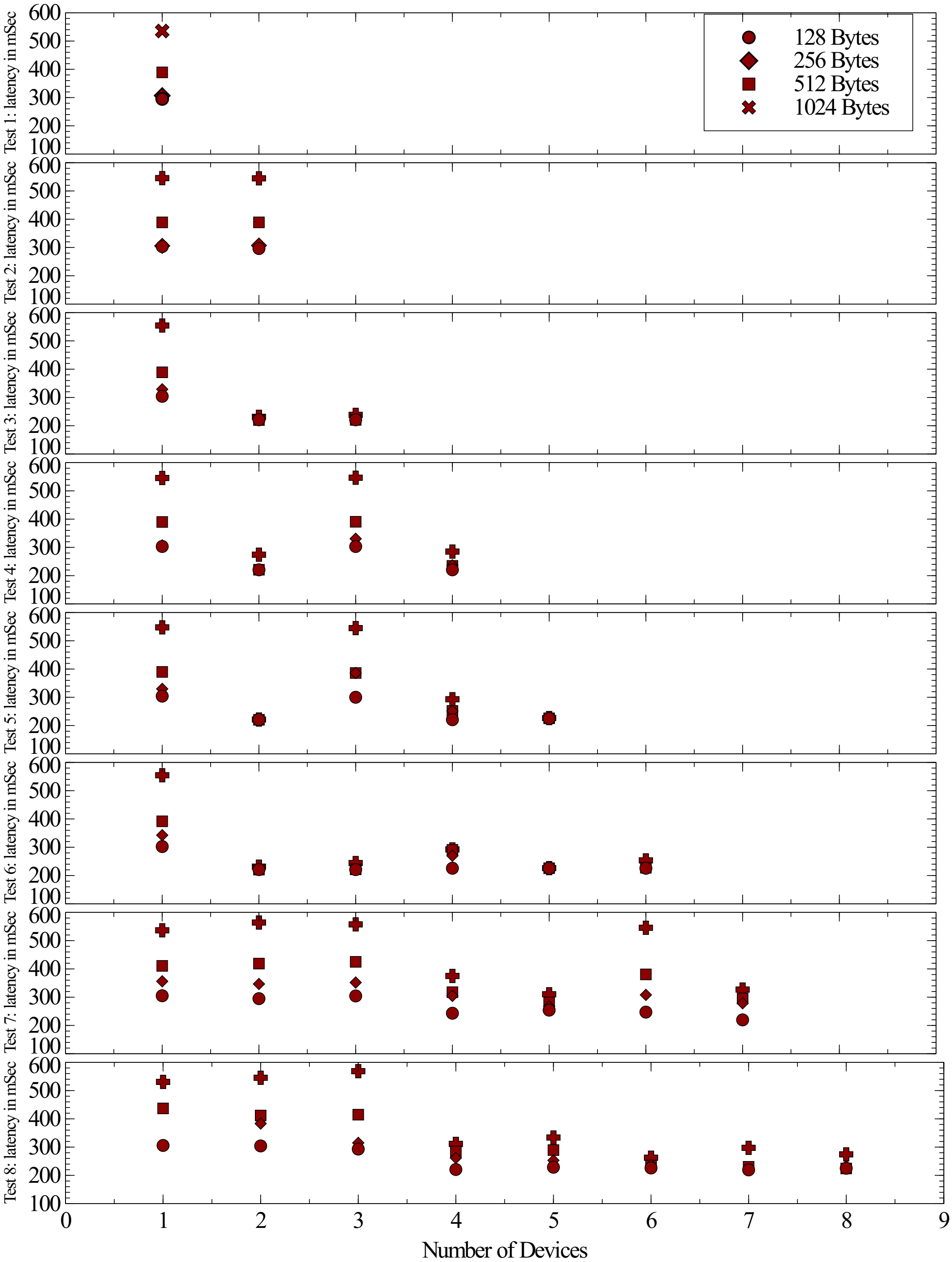


Figure 5.28: Latency LTE-M

Measurement	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
PD	0	0	0	0	0	0	0	0
Number of Reconnections	0	0	0	0	0	0	0	0

Table 5.12: LTE-M network capacity test

by d2, whereas the lowest E2E value of 219 mSec was recorded by d7.

Finally, in test 8, devices d1, d2, and d3 had higher E2E values than devices d4 to d8. The highest E2E value of 568 mSec was recorded by d3, whereas the lowest value of 219 mSec was recorded by d7.

Figure 5.29 shows the bitrate performance of LTE-M across all the network capacity tests. The X-axis represents the number of connected devices, whereas the Y-axis represents the calculated median bitrate for each device.

In the case of test 1, the highest calculated bitrate of 14.49 Kbps whereas the lowest bitrate is 3.28 Kbps.

For test 2, the highest bitrate of 14.52 Kbps and 3.16 Kbps is calculated for device d1.

In the case of test 3, the highest bitrate of 33.709 Kbps is calculated for device d2, whereas the lowest bitrate of 3,17 is calculated for device d1.

In the case of test 4, device d2 has the highest calculated bitrate value of 27.11 Kbps whereas device d3 has the lowest bitrate of 3.16 Kbps.

During test 5, the highest bitrate of 34.114 Kbps was calculated for device d2, and the lowest bitrate of 3.16 Kbps was calculated for device d1.

In the case of test 6, the highest bitrate value of 34.13 Kbps was calculated for device d6, whereas the lowest value of 3.16 Kbps was calculated for device d1.

For test 7, the highest calculated value of 25.121 Kbps was calculated by d5, whereas the lowest value of 3.16 was calculated by devices d1, d2, and d3.

Similarly, for test 8, the highest bitrate of 29.68 Kbps was calculated for device d6, whereas the lowest bitrate of 3.16 Kbps was calculated for device d1, d2, and d3.

Table 5.12 shows the PD and number of re-connections recorded during the LTE-M capacity test. As can be seen from the table, there were no PDs as well as network reconnections recorded during the entire test.

Figure 5.30 shows the E2E latency calculated during the NB-IoT network capacity testing. The X-axis represents the number of connected devices, whereas the Y-axis represents the calculated median E2E latency values during each test.

In the case of test 1, the lowest E2E latency was 1900 mSec, whereas the highest calculated E2E latency was 2102 mSec.

In test 2, the lowest E2E latency of 1900 mSec was calculated for device d1, whereas the highest E2E latency of 2880 mSec was calculated for device d2.

Similarly, for test 3, the lowest E2E latency of 838 mSec was calculated for device d1, whereas the highest E2E latency of 5257 mSec was calculated for device d3.

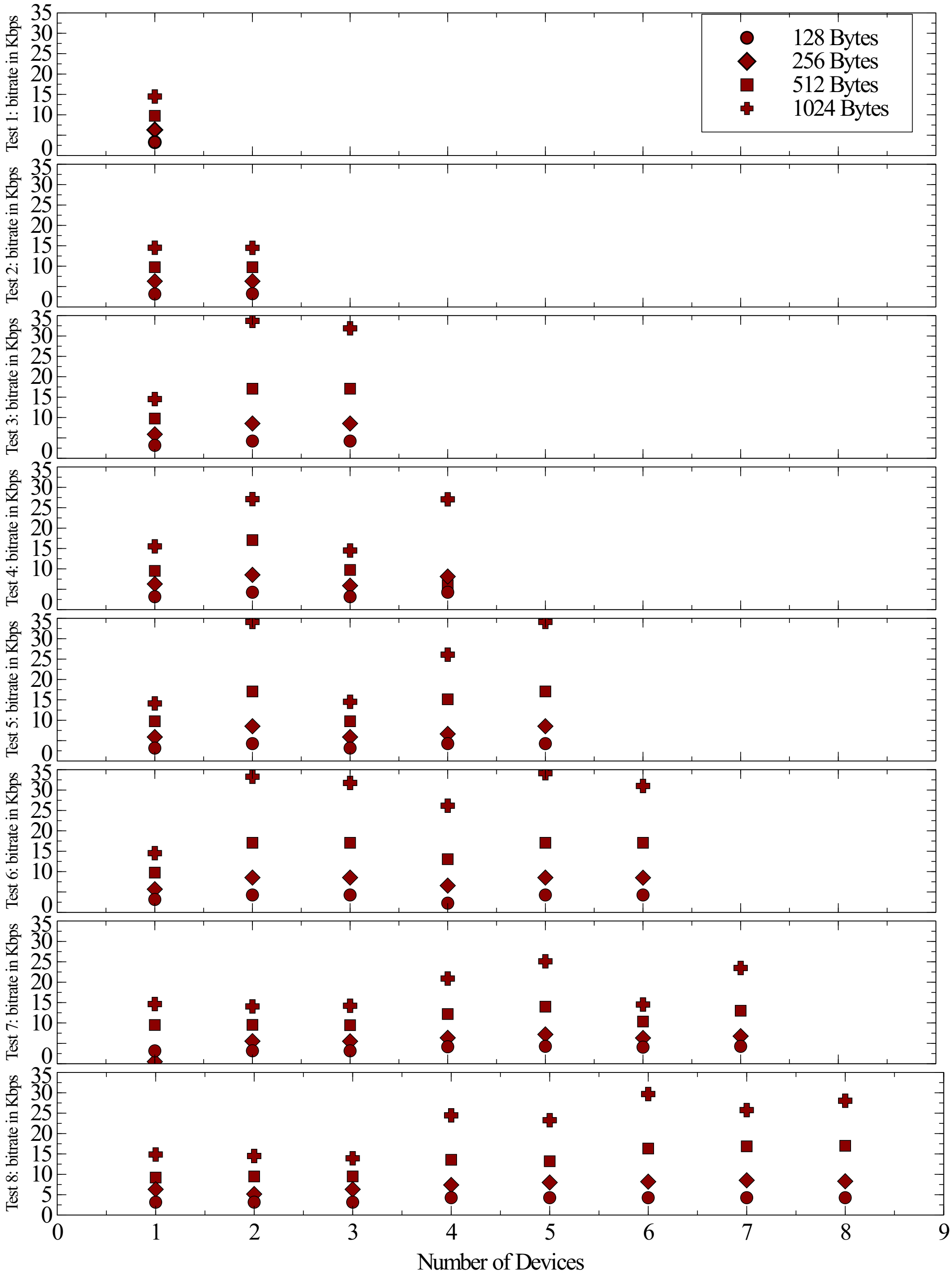


Figure 5.29: Bitrate LTE-M

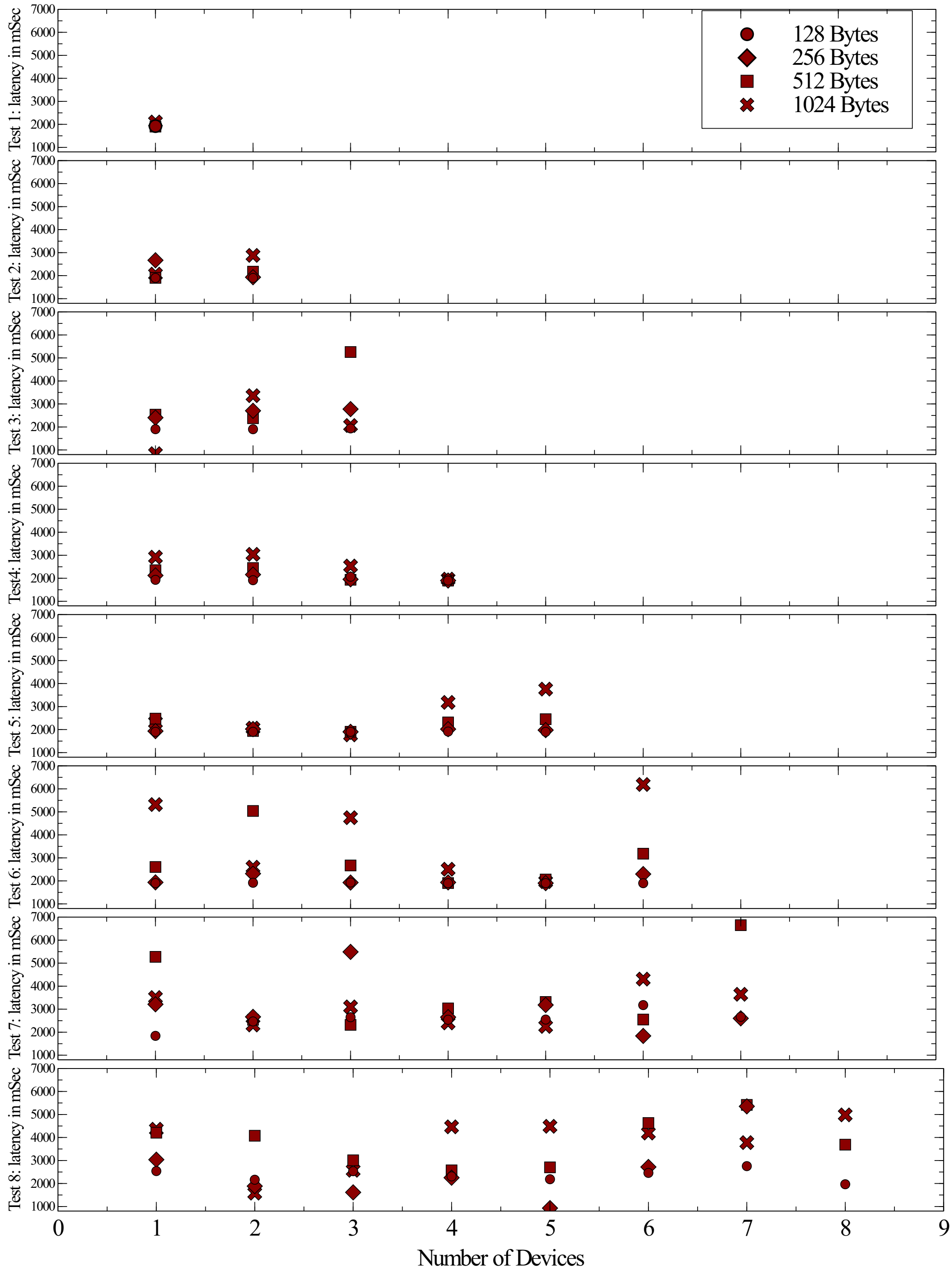


Figure 5.30: Latency NB-IoT

Measurement	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
PD	17	39	42	38	41	84	105	152
Number of Reconnections	6	15	18	16	20	44	56	58
Power Cycle	0	0	0	1	1	3	2	2

Table 5.13: NB-IoT capacity test

In the case of test 4, the lowest E2E latency of 1900 mSec was calculated for device d4, whereas the highest latency of 3040 mSec was calculated for device d2.

In test 5, the lowest E2E of 1768 mSec was calculated for device d3, whereas the highest E2E latency of 3758 mSec was calculated for device d5.

Similarly, in the case of test 6, the lowest E2E of 1900 mSec and the highest E2E latency of 6189 mSec was calculated for device d6.

In the case of test 7, the lowest E2E latency of 1841 mSec was calculated for device d1, whereas the highest E2E latency of 5490 mSec was calculated for device d3.

Finally, in the case of test 8, the device d2 had the lowest E2E latency of 1580 mSec, whereas the device d7 had the highest E2E latency of 5353 mSec.

Figure 5.31 shows the total calculated bitrate for all the devices during all the network capacity tests. The X-axis in the graph represents the total number of connected devices during the test, whereas the Y-axis represents the calculated bitrate for each device during different tests.

During test 1, the lowest calculated bitrate for device d1 was 0.53 Kbps whereas the highest calculated bitrate was 3.82 Kbps.

In test 2, device d1 and d2 had the same lowest bitrate of 0.533 Kbps whereas device d1 had the highest bitrate of 3.99 Kbps.

In the case of test 3, the lowest bitrate of 0.52 Kbos was calculated for device d2, whereas device d3 had the highest calculated bitrate of 3.99 Kbps.

Similarly, in test 4, the lowest bitrate of 0.52 Kbps was calculated for device d3, and the highest bitrate of 4.11 Kbps was calculated for device d4.

In the case of test 5 and test 6, all the test devices had the same lowest calculated bitrate of 0.53 Kbps. In contrast, device d3 had the highest calculated bitrate of 4.47 Kbps during test 5, and device d5 had the highest bitrate of 4.1 Kbps.

Similarly, for test 7, device d6 had the lowest calculated bitrate of 0.3 Kbps, whereas device d5 had the highest bitrate of 3.6 Kbps.

Finally, in the case of test 8, the lowest bitrate of 0.32 Kbps was calculated for device d7, whereas the highest bitrate of 5.24 Kbps was calculated for device d2.

Table 5.13 shows the PD as well as the number of network re-connections and power cycles recorded during the NB-IoT capacity testing. As can be seen from the table, the highest number of PD of 152 packets was recorded for all the devices in test 8. At the same time, the highest number of network re-connections were recorded during the same tests. As can be highlighted from the table, the number of dropped packets and the number of times a device performed a network re-connection increases as the number of test

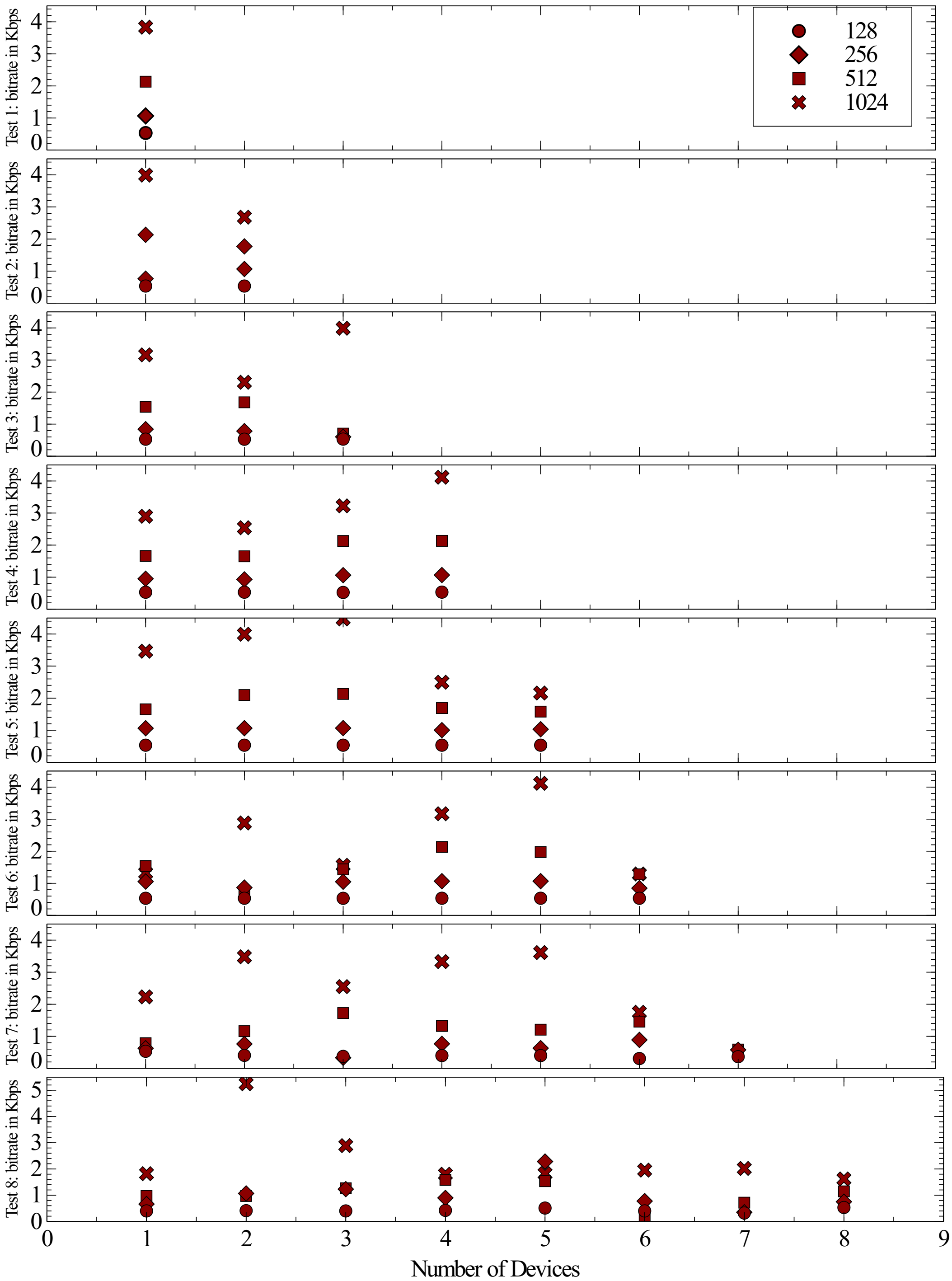


Figure 5.31: Bitrate NB-IoT

Test number	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Avg. test time (LTE-M)	4	4	3	3.5	2.8	2.2	3.14	3
Avg. test time (NB-IoT)	30	43.5	36	31.75	30.4	46.33	51.57	53,62

Table 5.14: C-IoT average test time in minutes

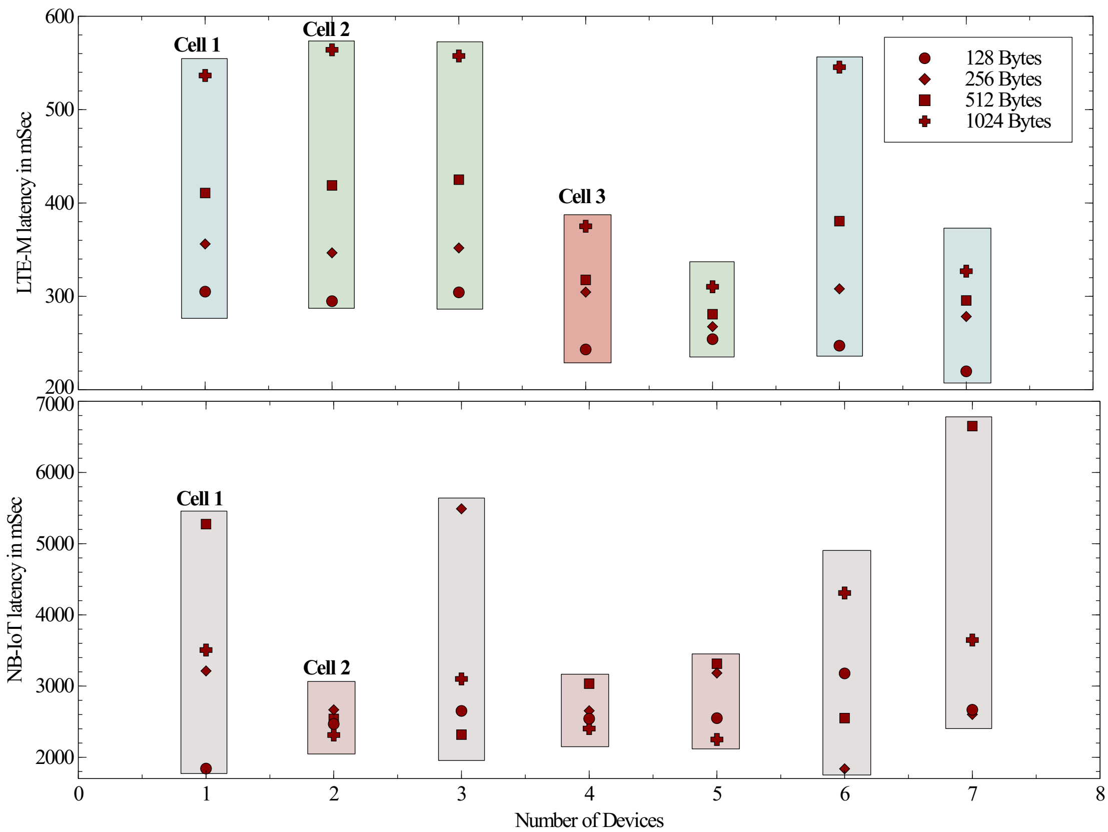


Figure 5.32: Performance difference between different cells

devices increases. Similarly, in the case of power cycles, the test devices in the network go on increasing, and the number of power cycles required for the device to reconnect and restart the test again also increases. This is a very contradicting result to what was observed in the case of LTE-M in similar test settings (shown in Table 5.12).

Table 5.14 shows each device's average time to complete the network capacity testing. The table shows that the maximum time taken by LTE-M devices in all the tests was 4 minutes. In the case of NB-IoT, a clear pattern can be observed as the number of connected devices increases and the average time to complete each test increases. The maximum time taken to complete the test in case of NB-IoT was recorded to be slightly over 53 minutes.

In order to further investigate different hypotheses, it was decided to check device metadata on the serving eNB id for all the devices and the signal level of all the devices during the test. The observations are as follows:

- In the case of LTE-M, it was discovered that the devices connected to three differ-

Measurement	Device 1	Device 2	Device 3	Device 5	Device 6	Device 7	Device 8
Packet Loss	9	4	4	3	0	0	3
Number of Reconnection	2	1	1	1	0	0	1

Table 5.15: LoS NB-IoT capacity test

ent cells during the test. This was interesting because the devices reported similar signal quality within 1 to 2 dBm margins in different measurements. Figure 5.32 highlights which devices were connected to which cells at the testing time. Devices d1, d6, and d7 are connected to Cell 1. Similarly, devices d2, d3, and d5 are connected to Cell 2. Finally, device d3 is connected to Cell 3.

- The performance of different test devices varies depending upon the cell it is connected.
- The similar finding was made during reviewing the data from NB-IoT capacity testing, and just like the LTE-M tests, the network performance is different for different devices based on which cell it is connected to. Figure 5.32 also highlights which NB-IoT devices are connected to which cell. In the case of NB-IoT, devices d1, d3, d6, and d7 are connected to Cell 1, whereas devices d2, d4, and d5 are connected to Cell 2. The performance gap between the two cells is more visible in this case.
- Another behavior observed in the case of NB-IoT capacity testing was that each device connected to the same cell as the previous test unless the device was power cycled. Once the device was power cycled, it was attached to a completely different cell.
- Another observation was that, in the case of LTE-M, the maximum number of devices connected to one cell was five. In contrast, in the case of NB-IoT, there were a maximum of four devices connected to each of the cells.

The cell selection was performed automatically by the IoT device; therefore, it was decided to redo the LTE-M and NB-IoT cell capacity tests after mid-night from a LoS location that is approximately 160 meters away from the cell tower. The considered test time was under the assumption that the C-IoT base station on DTU campus would be less loaded with regular mobile communication traffic during this time.

Figure 5.33 shows the measured performance from the LTE-M and NB-IoT testing when measured in LOS condition after midnight by connecting all the eight test devices at once. The tests were repeated several times for both the technologies, and the outcome was the same every time.

In the case of LTE-M, we could only test 6 simultaneous devices attached to one cell. Therefore as it can be observed from figure 5.33, in the case of LTE-M testing, there was no connection to devices d2 and d4 during the test. Although, after the first six devices finished testing, d2 and d4 could connect to the same cell and finish the test. The observed performance for d2 and d4 is similar to the other devices. No PD and network reconnections were recorded while LTE-M testing.

Similarly, in the case of NB-IoT, we could only test 7 simultaneous devices attached to one cell. This can be seen from the figure 5.33. Device d4 had no connection until the rest of the devices finished their testing. Unlike LTE-M, the NB-IoT testing resulted in many dropped packets and network reconnections.

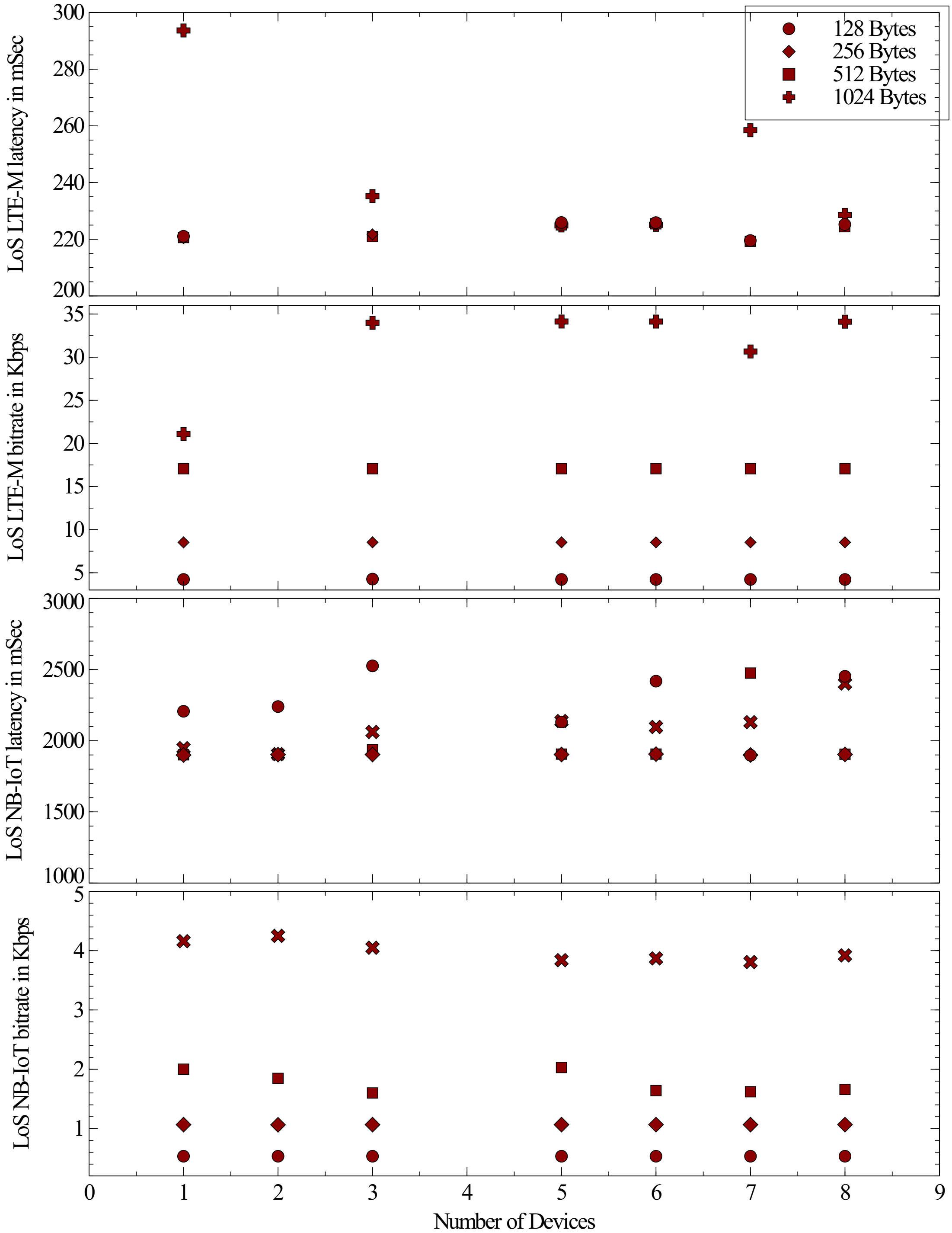


Figure 5.33: Line of sight C-IoT capacity tests

Table 5.15 shows the overview of PD and several reconnections experienced by different devices while testing in LoS with the cell.

As can be seen from different measurements performed during all the experiments, a few critical outcomes can be highlighted, such as:

- LTE-M performs much better than that of NB-IoT as we keep adding devices to the network.
- Another outcome from this test is that LTE-M performance depends on the number of cells available in a given area and how much traffic they carry from other wireless devices and technologies. The difference in the performance of different cells is visible in figure 5.32.
- In the case of LTE-M, the maximum number of six simultaneous devices indicates that the operator has implemented the minimum requirements for 6 Physical Resource Block (PRB) stated in LTE-M standard (discussed in 2.2.3. This is the minimum bandwidth requirements from the LTE-M standards.
- In the case of NB-IoT, the network performance started degrading as we added new devices during the test. The device needed to restart the network attach procedure several times during the test and, in some cases, needed to be power cycled.
- In the case of NB-IoT, once the device was power cycled, the device changed the cell it was initially connected. This might have been because lack of handover support in the NB-IoT standard. When one NB-IoT cell reached its maximum device capacity, the network could not hand over the device to another cell with more available resources.
- In the case of the LoS testing of NB-IoT, it was observed that each cell could support a maximum of 7 simultaneous devices that are attached to the network. The number of attached devices differs from the number of devices sending the packets during the test. When further investigating the exact timestamp of the received packets, it was discovered that, after starting the test in the first two minutes of testing, only 4 devices were sending the data. The other devices sent packets once these devices started sending higher payloads.
- The numbers observed during the test might vary between in-band and guard-band deployment of the NB-IoT network. A similar observation was made in the initial capacity testing of NB-IoT, that each cell accepted a maximum of 4 devices. This indicates the operator might have deployed the NB-IoT in a guard band configuration and is not affected by the traffic patterns experienced by the cell.

5.6 IoT protocols and power consumption

One of the highlighting features of NB-IoT and LTE-M technologies is that they can support the deployment of devices with a battery life of up to 10 years with a 5-watt hour battery capacity [96].

Andre *et al.* [89] calculates the battery lifetime estimates of NB-IoT and LTE-M networks by performing modeling as well as experimental validation. Their findings showed that achieving up to 10 years of battery life is possible if the devices and the network configuration parameters are optimized. Haider *et al.* [101] performed a real-world evaluation of battery consumption and performance of NB-IoT in Malaysia and discovered that the NB-IoT battery life depends heavily on the signal quality and latency of the communication. When compared with Sigfox and LoRaWAN, NB-IoT was found to have the worst battery performance in similar measurements. Mads *et al.* [102] performs an experimental evaluation of first generation C-IoT modules using Keysight UXM eNB emulator and discovers that the battery life calculated by performing this study is only 10% shorter than that of 3GPP estimations. Pascal *et al.* [103] performs a modeling study to perform the power consumption analysis of NB-IoT and LTE-M in challenging, smart city environments. The results from the study reveal that LTE-M is 4% power efficient than that of NB-IoT in MCL 144 dBm to 154 dBm. However, even lower coverage levels NB-IoT show better battery life performance. Rudi *et al.* [104] performance comparison between PSM and eDRX modes offered by NB-IoT to find out which one can offer longer battery life in a flood monitoring system deployment. The tests were performed by sending data using HTTP protocol, and the results showed power consumption using PSM is 52% higher than that of eDRX. Chunhe *et al.* [105] performs an experimental evaluation of power consumption of NB-IoT UE when exposed to complex electromagnetic interference. The tests included sending UDP payload to a remote server using two different UEs, one with cell reselection parameter ON and the second with cell reselection parameter OFF. It was discovered that when the NB-IoT UE is performing the network attach, and it can take a longer time to perform the cell selection procedure when deployed in an area covered with multiple NB-IoT eNBs. The time taken to perform the cell selection directly affects the device's power consumption. Svetoslav *et al.* [106] performs an experimental evaluation of power consumption experienced by a NB-IoT UE when sending data using three different NB-IoT networks in Germany. The results highlighted performance differences in different NB-IoT networks, and with network and peripheral optimization, it is possible to achieve 10 years of battery life.

One can notice from the current research outcomes that different parameters such as NB-IoT UE, use of PSM and eDRX features, MNO networks, coverage levels at the deployment locations, etc. seems to affect the power consumption and the expected battery life of the C-IoT network. Although these results are very insightful, they lack the complete picture of the power consumption performance of C-IoT networks. This is in terms of the use of multiple kinds of UE, use of multiple MNO networks (including NB-IoT & LTE-M), and use of different transport protocols tested in different coverage conditions. Therefore it was decided to perform an experimental evaluation of the power consumption of the different communication modules used in this thesis using SaRa-R410 and Pycom Gpy [107]. The test was conducted by sending UDP and TCP packets using different MNO C-IoT networks in Denmark. The test setup used during the test consisted of

- The SaRa-R410 and Pycom Gpy modules were controlled by using AT commands to send and receive data.
- The remote server for sending the data was the same as all the other measurements. The server was hosted in Frankfurt, Germany.
- The tests were conducted on the Technical University of Denmark (DTU) campus from a location in LoS, NLoS, and deep indoor from the eNB present on campus.
- The tests were conducted in three locations, the first location had excellent signal quality > -75 dBm, the second location where the signal quality drops below < -100 dBm, and the third where the signal drops below < -110 dBm.
- The tests were repeated using the same operators OP1 and OP2 as the rest of the

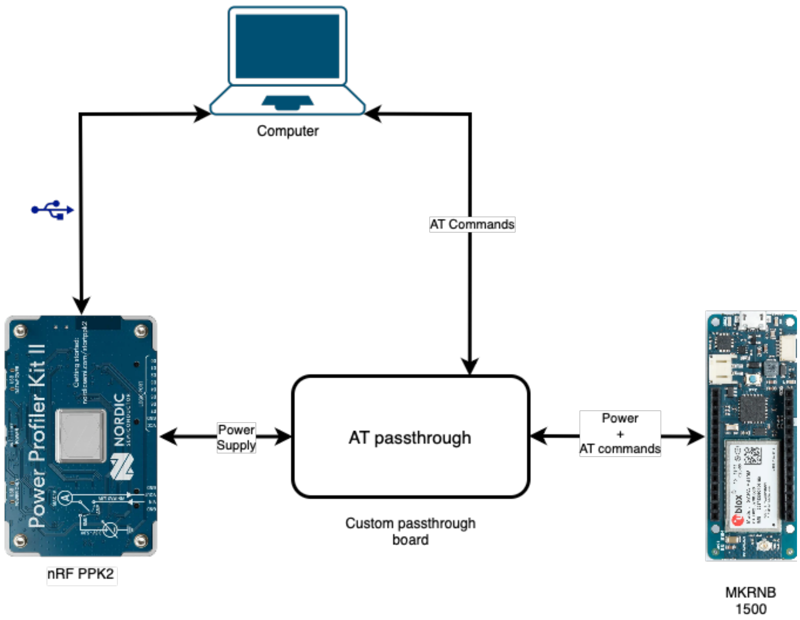


Figure 5.34: Power measurement setup

tests in this chapter.

- The tests were conducted using both NB-IoT and LTE-M networks provided by both the MNO.

Components used in the test setup:

Figure 5.34 shows the experimental setup contracted while performing the power consumption tests. The power measurement device used during the test is from Nordic Semiconductors called nRF Power Profiler Kit 2 (PPK2) [108]. The nRF PPK2 supports measurements of up to 1A current draw and has a very easy-to-use user interface to conduct the measurements. The nRF PPK2 allows using a sampling frequency of up to 100000 samples per second. This allows us to detect minor fluctuations in the power measurements. The device used to calculate the power consumption is Arduino MKRNB1500. The device uses the same microcontroller SAMD21 used in developing the final prototype during the PhD project. Similarly, MKR1500 uses the same U-Blox SARA-R410 module as the prototype device.

In order to conduct the measurements, the device needed to be connected to the nRF PPK2 and to the computer to give AT commands to the module. Therefore a custom board was developed that allowed us to provide the power through the nRF PPK2 while allowing the AT commands from the computer to pass through. The device power consumption tests are performed after the device has been connected to the network. Also, while performing all the tests, the antenna and the peripheral devices used were the same. Therefore the only two variables while testing was the choice of transport protocol (either UDP or TCP) used, coverage level, and the choice of the MNO. The results were obtained by repeating the tests several times over several days to check for irregular behavior in the network while sending the data. Some results obtained from the tests are highlighted in this section, and the rest are summarised in a table at the end.

Figure 5.35 shows the complete power consumption graph recorded by sending a UDP

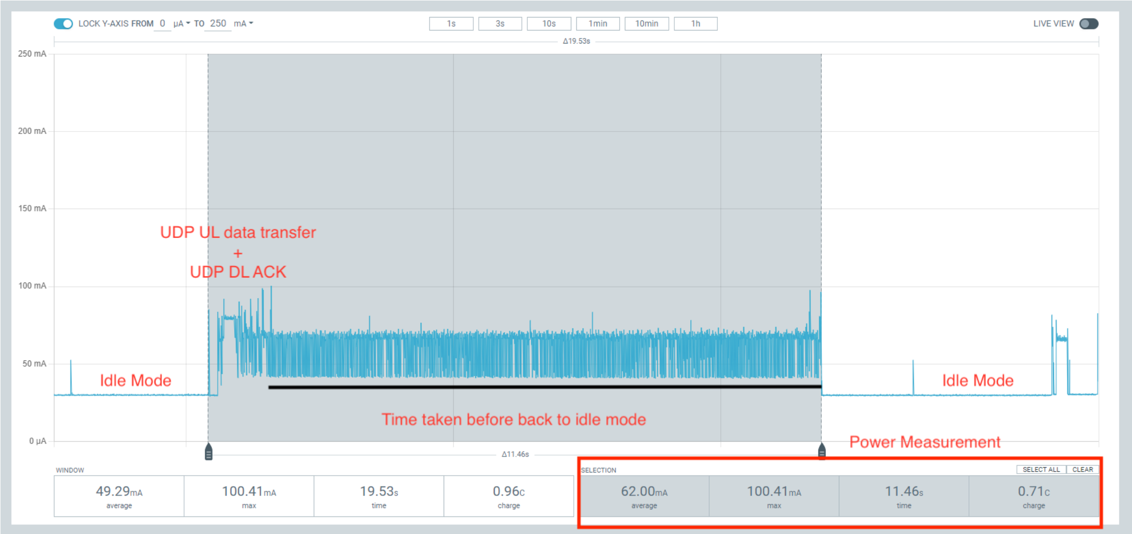


Figure 5.35: Example of NB-IoT UDP data transmission using OP2

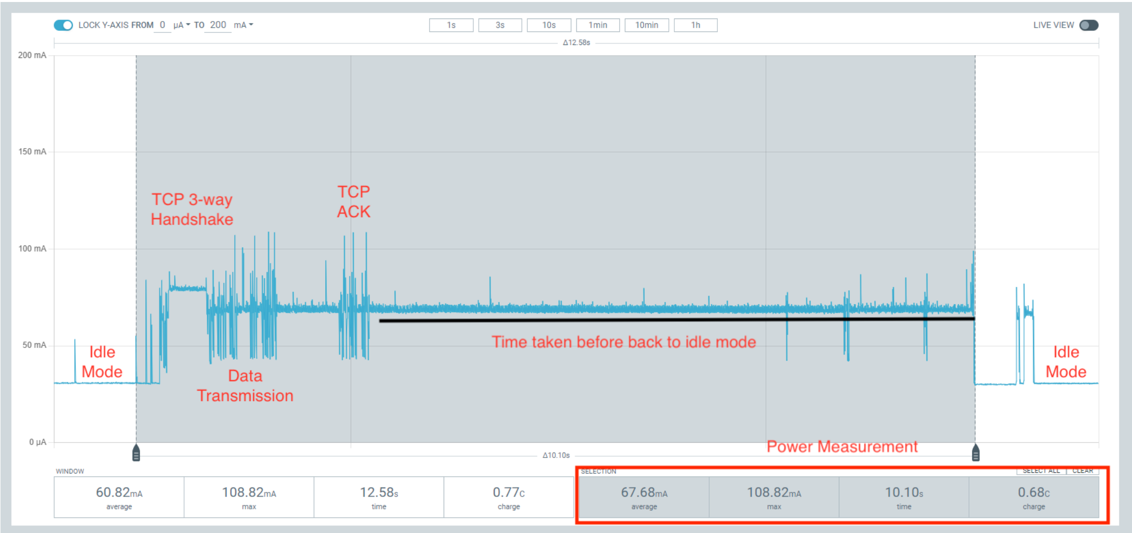


Figure 5.36: Example of NB-IoT TCP data transmission using OP1

packet using OP2. The device follows the following procedure when the UDP packet is sent:

- A UDP socket is OPEN. This brings the device's power consumption from idle mode to active mode.
- Once the socket is OPEN, the data in HEX format is written to the socket and transmitted to the remote server.
- The remote server sends a DL acknowledgment when the data is received.
- The device stays in active mode for 8 more seconds before returning to idle mode.

Similarly, figure 5.36 shows the power consumption recorded by sending a TCP packet using OP1. The device follows the following procedure when a TCP packet is sent:

- After the TCP packet is OPEN, the device changes from idle to active state and performs a TCP 3-way handshake.

Measurement for NB-IoT	OP 1 UDP			OP 1 TCP			OP 2 UDP			OP 2 TCP		
	RSRP >-100 dBm	-110 dBm <RSRP < -100 dBm	RSRP < -110 dBm	RSRP >-100 dBm	-110 dBm <RSRP < -100 dBm	RSRP < -110 dBm	**RSRP >-100 dBm**	-110 dBm <RSRP < -100 dBm	RSRP < -110 dBm	RSRP >-100 dBm	-110 dBm <RSRP < -100 dBm	**RSRP < -110 dBm**
Avg. Current (mA)	68.82	72.73	No connection	67.68	71.46	No connection	62	66.85	117.99	63.23	62.31	137.14
Peak Current (mA)	112.60	278.31	No connection	108.82	266.78	No connection	100.41	244.45	282.10	103.73	114.49	319.19
Transmission Time in Seconds	8.44	9.45	No connection	10.10	10.66	No connection	11.46	11.73	28.98	13.32	15.05	54.46

Table 5.16: NB-IoT power consumption tests

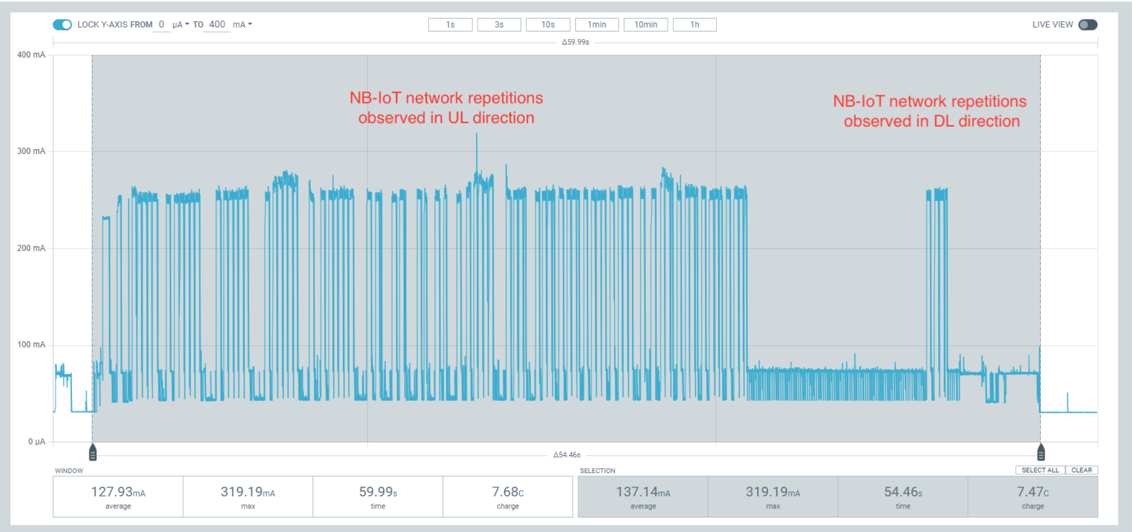


Figure 5.37: NB-IoT TCP data transmission in RSRP < -110 dBm

- As soon as the 3-way handshake is complete, the device sends a UL packet to the remote server and waits for the TCP acknowledgment.
- Once the TCP acknowledgment is received, the TCP socket is closed, and the device waits another 8 seconds before returning to idle mode.

Table 5.16 shows the average current, peak current, and the total transmission time for both OP1 and OP2 by sending UDP and TCP packets over NB-IoT. In the case of NB-IoT, OP1 has higher current consumption than that OP2 for both UDP and TCP data transfer. At the same time, OP2 takes longer to transfer both UDP and TCP packets. In the case of OP1, there were issues in sending the data at extremely low coverage levels. This was mainly because the device was rebooting once the device was attached and the data send command was issued. In the case of OP2, it was possible to measure the current consumption in extremely poor radio conditions. As can be seen from the table, the time taken to transmit the data transmission was 28.98 seconds and 54.46 seconds for UDP and TCP data transfer, respectively. The considerable time delay in sending the packets was due to repetitions observed in the NB-IoT network. This behavior of the NB-IoT network can be observed in figure 5.37.

Table 5.17 shows the average current, peak current, and total transmission time observed while performing the LTE-M power consumption tests using OP1 and OP2 in different radio conditions. As can be observed from the current consumption of OP2 is higher than that of OP1 in good radio conditions, whereas OP1 has higher current consumption in poor radio conditions than that of OP2. Similar to the NB-IoT testing, the device showed similar behavior in the extremely poor radio conditions for OP1. One hypothesis for device

Measurement for LTE-M	OP 1 UDP			OP 1 TCP			OP 2 UDP			OP 2 TCP		
	RSRP >-100 dBm	-110 dBm <RSRP < -100 dBm	RSRP < -110 dBm	RSRP >-100 dBm	-110 dBm <RSRP < -100 dBm	RSRP < -110 dBm	**RSRP >-100 dBm**	-110 dBm <RSRP < -100 dBm	RSRP < -110 dBm	RSRP >-100 dBm	-110 dBm <RSRP < -100 dBm	**RSRP < -110 dBm**
Avg. Current (mA)	54.11	58.62	No connection	60.54	65.39	No connection	106	112.77	108.29	108.95	111.34	114.72
Peak Current (mA)	148.98	333.74	No connection	146.61	336.59	No connection	180.07	225.53	308.87	195.97	234.37	354.73
Transmission Time in Seconds	7.055	8.11	No connection	7.790	9.686	No connection	6.52	6.647	8.770	7.33	8.599	10.52

Table 5.17: LTE-M power consumption tests

Measurement for LTE-M	OP 1 UDP		OP 1 TCP		OP 2 UDP		OP 2 TCP	
	MKRNB 1500	PYCOM GPY	MKRNB 1500	PYCOM GPY	MKRNB 1500	PYCOM GPY	MKRNB 1500	PYCOM GPY
Avg. Current (mA)	54.11	107.39	60.54	149	106	171.79	108.95	169.60
Peak Current (mA)	148.98	204.05	146.61	202.82	180.07	290.23	195.97	288.60
Transmission Time in Seconds	7.055	6.29	7.790	6.237	6.52	6.250	7.33	6.42

Table 5.18: LTE-M power consumption tests MKRNB 1500 vs Pycom gpy

reboot in the case could have been that the total power consumption jumped above 400 mA, and the onboard voltage regulator was not equipped to support such a high surge in current. This led to rebooting of the device every time the message transfer was attempted. OP2 overall had taken less time than that OP1 while sending the packets once the send command was issued to the module.

The average current consumption in idle state MKRNB 1500 was recorded to be 31.28 mA, whereas for Pycom Gpy was 63.74 mA. Considering the idle mode current draw, it was decided to compare the power consumption of both of these boards using the OP1 and OP2 in good radio condition (RSRP > -100dBm).

Table 5.18 shows the average current, peak current, and transmission time recorded while testing the performance of MKRNB 1500 and Pycom Gpy devices using LTE-M technology from OP1 and OP2. The current consumption values in the table include these boards' idle mode power consumption. Even after considering the idle mode current consumption, Pycom Gpy consumed much higher current than MKRNB 1500 during data transmission. At the same time, the transmission time Pycom Gpy takes is lower than that of MKRNB 1500.

Considering all the power measurement tests, a few observations can be made:

- In the case of C-IoT, the device's power consumption depends on the signal condition the device is operating under.
- C-IoT networks offered by different MNO have different power consumption when tested using the same end device and similar network conditions.
- Different C-IoT end devices have different power consumption when tested against each other under similar network conditions.
- TCP tends to consume more power and takes a longer time for transmission than UDP.
- Different MNO network configuration settings and protocol stack implementation on

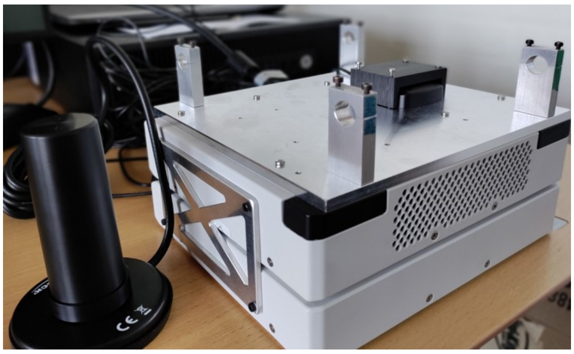


Figure 5.38: TSMA6B Radio Frequency scanner

devices contribute to different power performances of one particular protocol.

5.7 Other use cases for continuous monitoring system using C-IoT

During this project, there was an active collaboration with medical experts, doctors in different specializations, and patients. During one of the feedback sessions towards the end of the PhD project, it was discussed what other use cases, such a realtime heart monitoring system and, in general, C-IoT, could be used for.

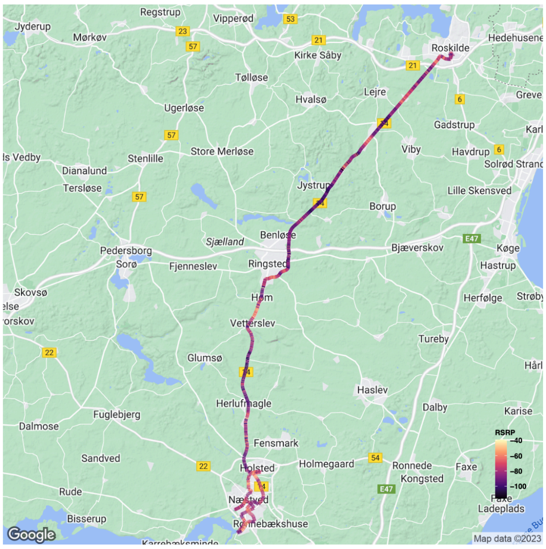
One such application was to monitor the patients transported to the hospital in case of an emergency. Although modern ambulances have very professional medical grade 12 lead ECG monitoring systems, these systems are often only capable of recording data locally. Therefore, paramedics must keep the hospital doctors updated about the patient's vitals. Therefore the intention was to see if the system developed during this PhD can be used for transmitting this data in realtime to the doctors who are waiting for the ambulances to arrive at the hospital so that they can be better prepared. The initial idea was to see if this system could be used to treat patients with Thrombosis [109].

In order to test the use of C-IoT in such a scenario, we needed to make sure there was adequate coverage of the technology so that the data could be transferred remotely when the vehicle was traveling at high speed.

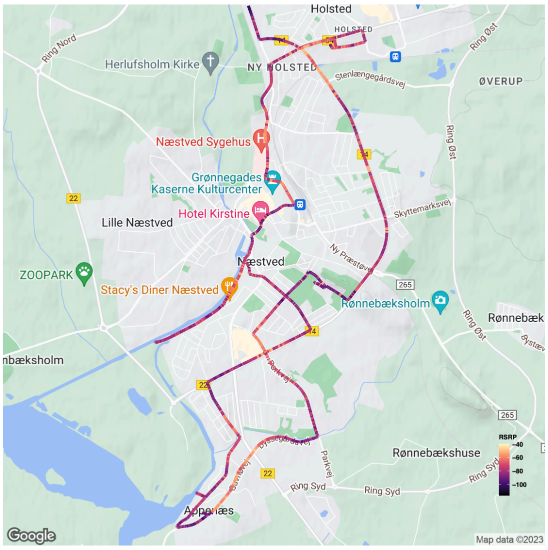
The first step in building such a system was to carry out some experiments to measure the C-IoT coverage around the paths taken by the ambulances. For this reason, we collaborated with Roskilde Hospital, Næstved region, and Ringsted region. The coverage of the C-IoT technologies was measured using TSMA6B Radio Frequency Scanner developed by Rohde & Schwarz [110].

Figure 5.38 shows the overview of the frequency scanner.

The TSMA6B RF Scanner supports scanning multiple cellular technologies deployed in the spectrum of 350MHz to 6 GHz. The TSMA6B scanner can record radio parameters cell id, Received Signal Strength Indicator (RSSI), Reference Signal Received Power (RSRP), Reference Signal Received Quality (RSRQ), Tx Power, etc., and cross-reference the measurements to a GPS location. All the data recorded by the scanner is then exported to a CSV file for further processing.



(a) LTE-M: RSRP Value Along the Route



(b) LTE-M: RSRP in Næstved City

Figure 5.39: LTE-M Coverage Measurement

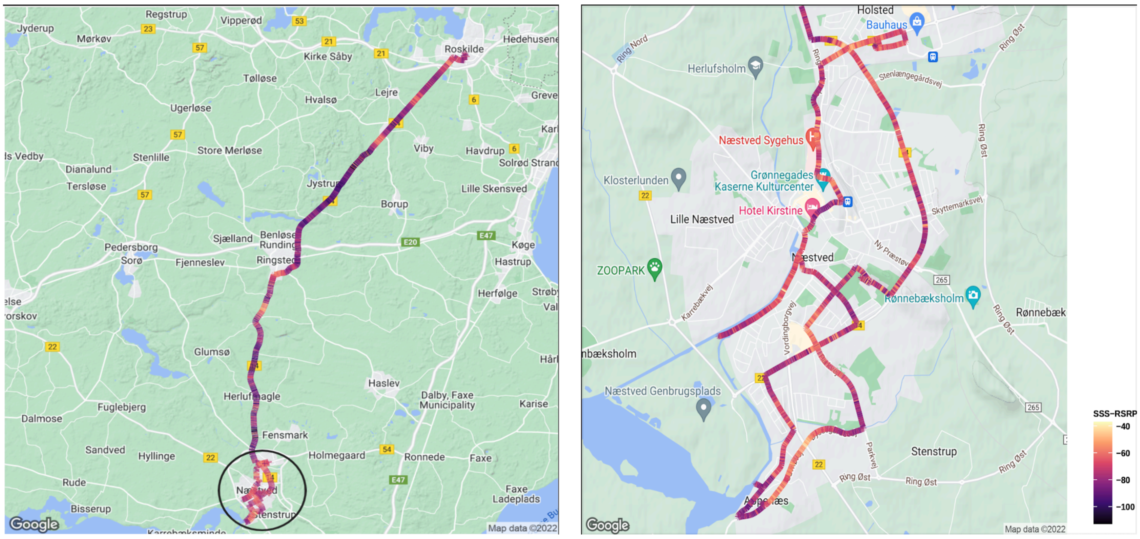
In order to perform the experiment, the TSMA6B scanner was mounted on top of an ambulance that was in service. This meant that we could record the network coverage along that path every time the ambulance was out on an emergency call. This was very interesting because we could capture data from a real incident.

In order to perform the experiment, the scanner was set to measure the performance of three technologies NB-IoT, LTE-M, and LTE. In the following section, the results obtained from each of the tests are discussed.

Figure 5.39 shows the coverage measurement of the LTE-M signal between Næstved City and Roskilde Hospital. The coverage RSRP values are recorded between -45 dBm to -107 dBm. Figure 5.39a shows the RSRP between Næstved city and Roskilde along the highway, whereas figure 5.39b shows the LTE-M coverage in the city of Næstved. As can be seen from the two different graphs, the LTE-M coverage in the city is much better than that of the highway.

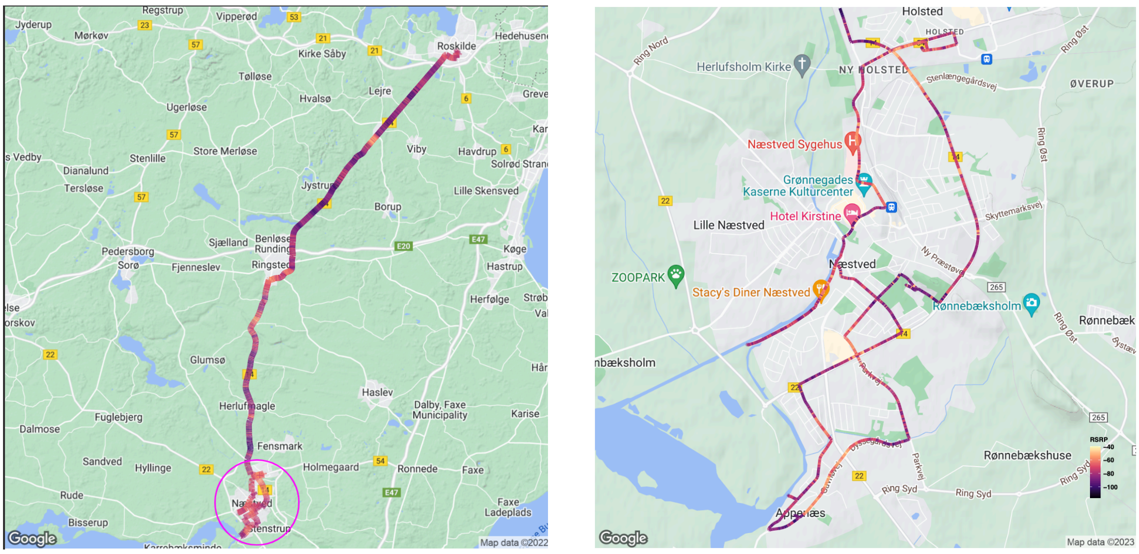
Figure 5.40 shows the observed NB-IoT coverage measurements during the tests. As seen from the different figures, NB-IoT has worse coverage than that LTE-M. This is especially evident along the highway as can be seen from figure 5.40a, the NB-IoT coverage in the city of Næstved is better seen from figure 5.40b. The range of RSRP measurements for NB-IoT is between -40 dBm to - 100 dBm. Given that NB-IoT technology does not support connection handover functionality at the time of testing. Therefore the technology has minimal use for this application, given that the ambulances on call are expected to move at a very high speed.

Figure 5.41 shows the LTE coverage measurements during the measurement. Figure 5.41a shows the RSRP measurement values for the highway measurements between Næstved city and Roskilde hospital, whereas figure 5.41b shows the LTE coverage in the Næstved city. The RSRP values recorded during these measurements ranged from -58 dBm to -120 dBm. When observed closely, it was discovered that the LTE-M performance during this measurement was worse than that of the LTE performance. This was especially surprising given that LTE-M coverage performance is expected to be better than that



(a) NB-IoT: RSRP Value Along the Route (b) NB-IoT: RSRP in Næstved City

Figure 5.40: NB-IoT Coverage Measurement



(a) LTE: RSRP Value Along the Route (b) LTE: RSRP in Næstved City

Figure 5.41: LTE Coverage Measurement

of LTE.

After further investigation into the scanning data and looking at the deployment of the eNB deployments around the test path, a probable hypothesis was formed.

- In Denmark, three operators have the radio network deployed on various technologies, but the nationwide C-IoT coverage deployment is only provided by two operators. Therefore, it was discovered that when the scanner was measuring the LTE RSRP performance, the results were based on the coverage from three operators in the region. However, when the scanner measured the C-IoT coverage, especially the LTE-M coverage, the measurements only included coverage from two operators. Therefore, the difference in the coverage could be because of the lack of MNO providing LTE-M coverage.
- At the time of measurements, the TSMA6B scanner was set to scan for the different technologies and not the specific frequency band. The TSMA6B scanner is capable of scanning between the spectrum of 350MHz to 6 GHz. When performing these measurements, the C-IoT is only deployed in band 20 (800 MHz) in Denmark, whereas LTE is deployed in multiple bands. This could be responsible for further improving the coverage of LTE over LTE-M.

5.8 Summery

This chapter measures the C-IoT network KPI by evaluating experiments in indoor, deep indoor, outdoor, remote outdoor, and in roaming scenarios. The C-IoT networks were further tested to determine the number of devices that can simultaneously be connected to a single eNB in a given area. The power consumption of different C-IoT communication modules was evaluated by sending TCP and UDP packets to a remote server using C-IoT technologies. Furthermore, in order to find alternate use cases for the use of a continuous ECG and HR monitoring system, the use of C-IoT was evaluated by measuring the coverage of C-IoT networks along the highways between Næstved City and Roskilde Hospital.

The tests highlight that the performance of the C-IoT networks in an indoor and an outdoor scenario is stable with fewer observed variations in the calculated network KPI parameters. In the case of the indoor tests, OP1 LTE-M and OP2 NB-IoT has experienced the most variation in the E2E latency performance. Overall, the C-IoT indoor tests indicate that OP2 NB-IoT has the worst performance compared to the rest of the tests. Similar observation can be made from the outdoor C-IoT KPI tests, that the OP2 NB-IoT network is subpar with the rest of the C-IoT networks.

The C-IoT, especially the LTE-M network, suffers degradation in the measured KPI when testing in deep indoor and deep-outdoor conditions. In many locations where there was no adequate LTE-M coverage, it was discovered that we could find coverage for other cellular communication technologies such as 2G and LTE. This was identified by performing network tests using an R&S TSM6B radio frequency scanner. The scanner also highlighted the absence of band 20 in deep indoor and deep-outdoor locations, further supporting the findings from the KPI testing. Coverage from 2G was present in almost all the locations, whereas LTE coverage was observed in limited locations while performing the deep indoor tests. Similarly, for the remote outdoor test, the scanner only reported the presence of 2G coverage, and no LTE coverage was observed. The absence of band 20 in such remote locations highlights the limitations of using only a single frequency band to deploy the C-IoT networks. Another observation from the remote outdoor testing of the C-IoT networks was the extended network attach times experienced by the test devices. In most cases, NB-IoT experienced more than 5 minutes delay before the device could find the correct network to attach. This issue was more prominent in the NB-IoT network deployed by OP2. One reason for such an extended network attach delay might have been the operator settings stored onto the SIM card [111]. Long delays in a network connection may lead to degradation in the battery life of a C-IoT device.

When the C-IoT networks were tested after three months intervals, there were some noticeable changes in the overall KPI performances across different MNO networks. The noticeable finding during this test was the absence of LTE-M coverage for OP1 during the deep indoor and outdoor tests. A closer look into the test results revealed that none of the C-IoT networks had coverage at point 1 while performing the outdoor KPI tests. After performing the tests a few days apart, the tests successfully highlighted a very peculiar behavior of the C-IoT networks and how a network outage scenario can affect the deployed IoT devices. This presents a challenge, when critical IoT applications that are realing on two MNO, one as primary and other as secondary to increase the reliability of the communication link.

While performing the network roaming tests, it was observed that the performance of KPI C-IoT is different in different countries even though the devices are roaming in the same MNO network. In our testing, we discovered that the home network NB-IoT KPI performance of the MNO is much worse than that of the roaming networks. Using different RAN vendors and misaligned radio parameters could be responsible for such a big difference in the network KPI. These differences can affect application's performance, especially the device's battery performance, if these differences are not considered while developing an IoT device. Another important finding from these roaming tests was the roaming architecture from different MNO. OP1 has followed the traditional roaming implementation where the device's internet traffic exits from the Danish packet gateway. In contrast, in the case of OP2, the data is pooled into one country instead of different packet gateways for different countries. This behavior was identified by capturing incoming packets on the remote server and checking the source IP for different packets.

Network capacity tests for continuous data transmission highlight that the cell capacity of LTE-M networks per sector is six devices when attached to an unloaded cell. In the case of NB-IoT, it is seven devices, although only four can experience a stable network performance. The cell capacity of LTE-M reduces to only three devices per cell when tested in peak traffic conditions, but for NB-IoT, it remained the same. This strongly suggests that the NB-IoT is deployed in a guard-band configuration for the tested MNO and cell capacity of guard-band deployment for NB-IoT is not affected by the cell traffic. NB-IoT tests show exponential degradation in performance as the number of connected devices increases. The cell capacity results indicate that the performance of the LTE-M networks is directly affected by the available resources in a given cell sector, and C-IoT might not have any reserved radio resources to guarantee the performance.

Power consumption tests show that the peak current consumption while transmitting data using different MNO results in different power consumption for the same end device. It was also highlighted that different IoT end devices consume different power when tested on the same MNO network under similar network conditions. The tests highlight the vast differences in the implementation of C-IoT networks by different MNO and RAN vendors, as well as the manufacturers of the IoT chipsets.

Similar to the observations from the C-IoT KPI tests, the LTE-M performance degrades as the number of eNB in a given area is reduced. This was observed while testing the C-IoT coverage along the highways between Næstved and Roskilde using an R&S TSM6B radio frequency scanner.

LTE-M is the primary communication technology used in transferring ECG and HR monitoring data from the patients in the application designed during this project. Therefore there is a need to find alternative ways of ensuring an active communication link between the end device and the remote server.

6 Future Directions

6.1 Improve reliability of Data transfer

As it was highlighted in the Chapter 5 at many test locations the performance of the Cellular Internet of Things (C-IoT) networks was not consist. This was especially evident in poor radio conditions. In several discussions with the medical professionals it was highlighted that how important is for the device to maintain some sort of communication with the backend systems. This was specially important given that the one of the motivation factor behind developing this system was to empower patients by presenting them with summery of their health parameters and to improve the Quality of Life (QoL) post Cardiovascular Disease (CVD). Therefore in this section we focused on developing and incorporating different techniques to improve the reliability of data transfer from the prototype device. The overall goal for this section was to maintain some sort of communication link with the backend systems if the primary communication technology fails. In this case, the primary communication technology was Long-Term Evolution Machine Type Communication (LTE-M). In order to improve the reliability of the communication following two methods were implemented.

- Use of Multiple Radio Access Technology (Multi-RAT)
- Use of Device-TO-Device (D2D) communication

In following subsections we will go over both the methods, the implementations, and performance.

6.1.1 Use of Multiple Radio Access Technology (Multi-RAT)

The use of Multiple Radio Access Technology (Multi-RAT) means using more than one Radio Access Technology (RAT) to send the information. In the case of this project, there were two technologies were chosen to send the data. The first technology was NarrowBand-Internet of Things (NB-IoT), a C-IoT technology, and the second was Long Range Wide Area Network (LoRaWAN). The reason for choosing both of these technologies was that both of these technologies have nationwide coverage in Denmark.

In the case of NB-IoT, the communication module used in the prototype device Ublox SARA-R412 supports it. Therefore, updating the device firmware to use NB-IoT when the primary communication technology (LTE-M) is unavailable was just a matter of updating the device firmware.

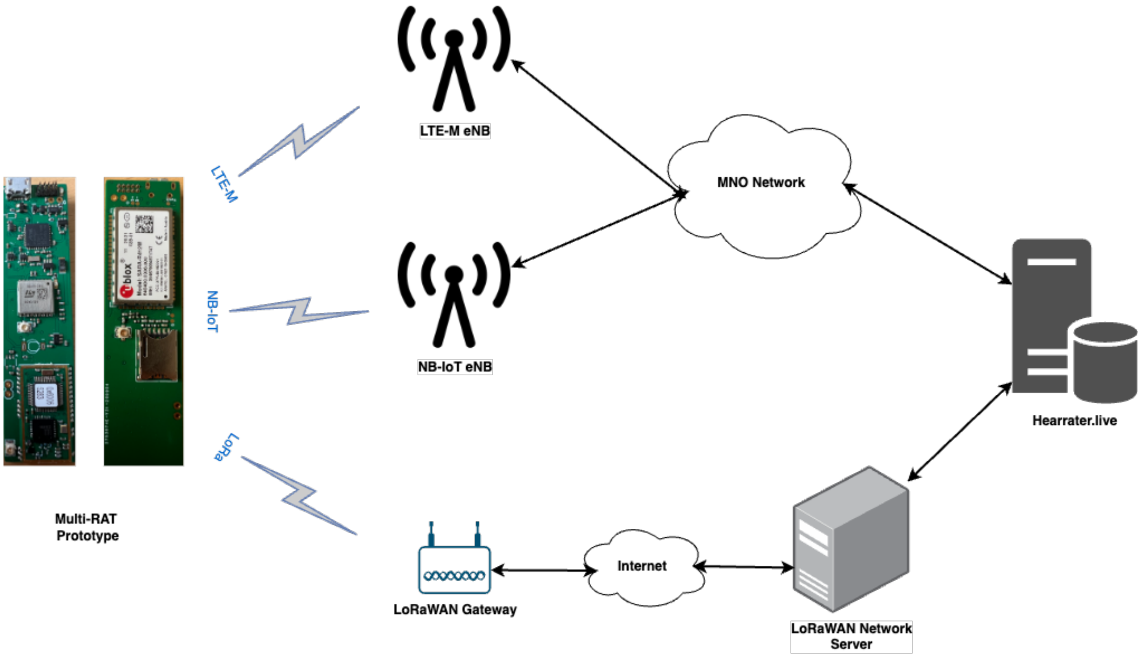


Figure 6.1: Multi-RAT Network Architecture

Figure 6.1 shows the Multi-RAT Network Architecture. As it can be seen from the diagram, there was a new prototype developed using the same components as that of the final end-device prototype in section 3.5. The only changes were that the BLE module Ublox NiNA-B312 was not added to this prototype version, and NEMEUS-MM002 communication module was added. NEMEUS-MM002 supports LoRaWAN and Sigfox communication technologies and can be controlled using AT commands. The NEMEUS-MM002 was connected to the SAM-D21 microcontroller using URAT communication links. Due to time constraints, the complete functioning of the complete Multi-RAT system was not performed. Instead working of each technology was verified individually.

6.1.2 Device-TO-Device (D2D) communication

Device-TO-Device (D2D) communication is where the end device or the User Equipment (UE) can communicate with another UE directly, without or with minimal involvement from the core of the cellular network. Widespread use of D2D communication happens in the emergency networks such as Terrestrial Trunked Radio (TETRA) [112]. There might be situations in which the RAT might not be accessible due to failures in a network. The unavailability of cellular networks affects various users. However, it is especially difficult for C-IoT devices, given that primary deployment of the C-IoT technologies occurs in a single frequency band. In the case of Denmark, all the Mobile Network Operators (MNO) have deployed their LTE-M and NB-IoT networks on band 20 (800 MHz). This means these devices will not be able to hand over to another band in case of network failure. In order to mitigate such scenarios, 3rd Generation Partnership Project (3GPP) first talked about D2D communication for Internet of Things (IoT) in release 12 by releasing a Proximity Service (ProSe) architecture [113].

Figure 6.2 shows the ProSe architecture described in 3GPP release 12. The architecture introduces a new Evolved Packet Core (EPC) entity called the ProSe function as well as five new interfaces (PC1-PC5) to connect different network elements. As per 3GPP

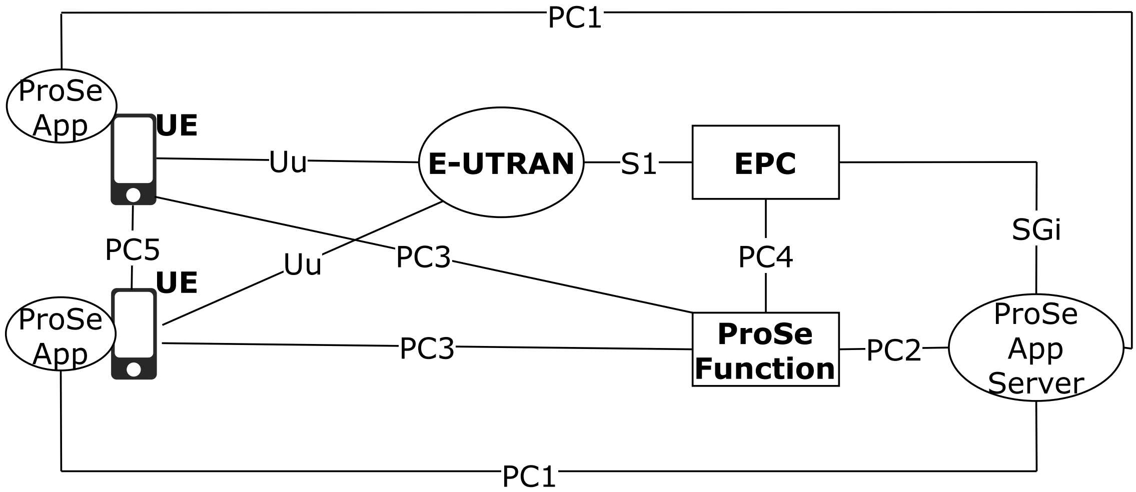


Figure 6.2: Proximity Service (ProSe) Architecture with LTE [113]

release 12, there are three scenarios where this ProSe architecture can be used. These three scenarios are described in Figure 6.3.

- **In Coverage:** In this scenario, the D2D communication can increase the total speed of data transfer from the network to a particular UE. This means if UE1 is receiving information from Evolved Node B (eNB), the path between eNB » UE2 » UE1 can be used for sending more information simultaneously, effectively increasing the data speeds for UE1.
- **Out of Coverage:** This scenario can mainly be used in emergency cases where the network infrastructure is unavailable for communication. In this case, UE1 and UE2 can communicate with each other directly using their own set of wireless frequencies. In this scenario, the UE can use both licensed as well as unlicensed frequencies.
- **Partial Coverage:** In this scenario, the UE, which is in coverage, can be used as a relay to extend the network's coverage. In this case, UE1 acts as a relay to extend the eNB coverage to UE2, which otherwise is out of coverage. Therefore, UE1 can forward the data from UE2 to eNB. This scenario significantly affects the UE's battery life connected to the eNB.

In release 14 from 3GPP, new work started on extending the D2D functionality to Cellular IoT devices. The work focused on simplifying the D2D channels and interfaces for NB-IoT and LTE-M for Machine Type Communication (MTC) applications [114]. Unfortunately, the efforts of using D2D MTC communication have been shifted to applications of D2D in vehicular communication. This implies that the work started in 3GPP release 14 of enabling the use of D2D in MTC applications has yet to be standardized to the best of our understanding from the 3GPP releases [115].

As observed in the deep-indoor testing of C-IoT networks, there were several locations in the underground tunneling system where no coverage was available from NB-IoT and LTE-M. Therefore, to address such situations, it was decided to continue working with the D2D concept and develop a hybrid approach using Multi-RAT and D2D communication.

At the time of implementation of D2D in the project, there were no commercially available C-IoT modules that supported the use of ProSe architecture proposed in 3GPP standards.

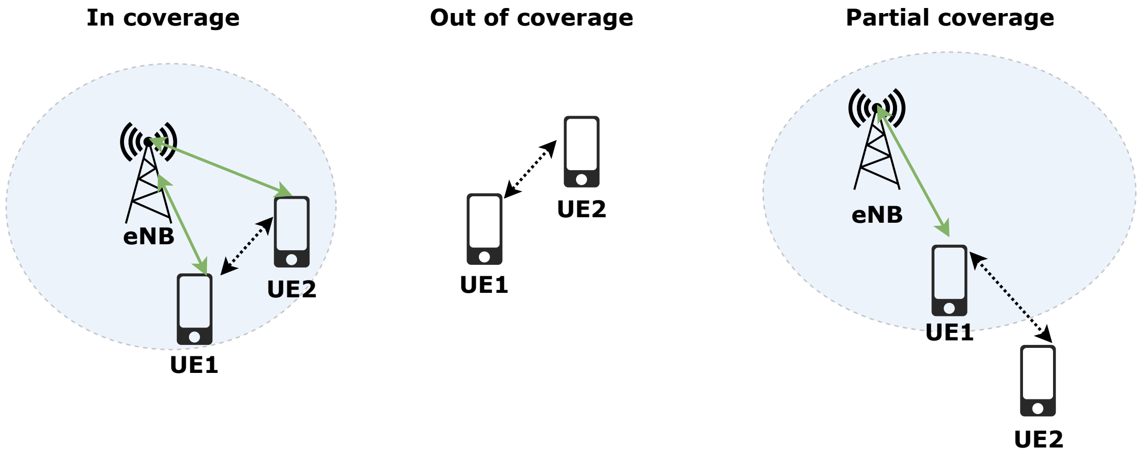


Figure 6.3: Deployment scenarios supported by ProSe architecture [113]

Therefore, to avoid a complete loss of connectivity, it was decided to use Long Range (LoRa) Chip to Chip communication to develop a working D2D setup.

6.1.3 Using Long Range (LoRa) and Long Range Wide Area Network (LoRaWAN) for Device-TO-Device (D2D) Communication

Konstantin *et al.* [116] highlights the benefits and drawbacks of implementing D2D communication over LoRaWAN and furhter go on to implement network assisted D2D communication protocol for commercial available LoRaWAN devices. Akram *et al.* [117] demonstrates use of LoRa D2D in transferring image sensor data in a forest environment. The authors further go on to implement a new scheme to overcome the bandwidth limitations of LoRa D2D communication. Jaehyu *et al.* [118] develops a method to secure D2D communication link between two LoRa devices. This is achieved by sharing cryptographic keys between the two devices that are used for manual authentication, confidentiality, and integrity of the communication. Nazmus *et al.* [119] have presented latency analysis for indoor applications of LoRa D2D communication. The experimental study is performed by varying the LoRa Spreading Factor (SF) and the communication bandwidth. The results show that, in case of LoRa D2D communication the lowest latency is observed with SF 7 and bandwidth of 500 kHz. Luca *et al.* [120] in their work try to use LoRa D2D to overcome the challenges presented by WiFi and cellular technologies in case of the emergency scenarios. A proposed solution, A LoRa-based Mobile Emergency Management System (LOCATE) which includes a mobile app interface, can be used to make short emergency communication over long distances. Alves *et al.* [121] evaluates the performance and energy efficiency of LoRa D2D. The authors go on to propose a network coded co-operations (NCC) where LoRa capable of sending combinations of multiple frames without affecting energy efficiency of the entire system.

The research related to use of LoRa for D2D indicates that the technology can be used for situations where the primary device is outside of regular network coverage. Most of the implementations of LoRa D2D focused on deploying the setup in a confined geographical location. The application developed in this project wanted to achieve exactly opposite of that, therefore we focused on developing a mechanism that can distribute the LoRa D2D communication frequencies dynamically based on the current location of the end device. The approach described in this section could help in implementing a nationwide, dynamic LoRa D2D network that can be used by the critical data delivery applications in case of complete loss of primary communication technology. The LoRa D2D channel, in this context is only used for sending a ”Keep-alive” messages, to ensure the bare minimum communication with the remote servers.

This section gives an overview of the LoRa D2D setup, which was developed to mitigate the complete loss of connectivity in deep indoor locations. In section 5.2.2, there were several points in the deep indoor coverage where no LTE-M or NB-IoT coverage was found. To avoid complete loss of data in such corner cases, it was decided to develop a D2D system by combining LoRa and LoRaWAN technologies and build an experimental setup based on the partial coverage scenario.

Figure 6.4 shows the end-to-end D2D experimental architecture using C-IoT as the primary communication technology and LoRa/LoRaWAN as the fallback D2D communication technology. The setup mainly consists of two communication channels; when the device is in good radio conditions, it will send data using the C-IoT. When the device is in an outage scenario, it will send data using the fallback D2D setup.

When the device is in good radio conditions and is powered ON for the first time, it performs the initial D2D setup shown in Figure 6.5.

The initial D2D setup procedure involves accessing available LoRa frequencies in the surrounding area.

1. End Device uses Representational State Transfer (REST) Application Programming Interface (API) provided by the application server to send its geolocation coordinates.

2. The application server uses these coordinates to compile a list of available LoRa D2D devices and their default channel frequency to the end device as a REST response. The list of available frequencies can be controlled by defining the radius parameter in the application server.

3. If the end device changes its location, it can send updated location coordinates to the application server to get the new list of available D2D frequencies.

Figure 6.6 shows the Interaction between the application server and end device using REST APIs. The device is using LTE-M to send and receive these JSON messages. The application server’s response is stored in a file called ”jsondata” on the communication module SARA-R412. This file is updated every time a new list of frequencies is available.

Figure 6.7 shows the physical locations on which the LoRa Device-TO-Device (D2D) devices are deployed. In this case, the IoT device is the one that has no C-IoT connectivity and is powered by an internal battery. LoRa Device-TO-Device (D2D) device runs on external power and always listening incoming messages on its default frequency. The location of the IoT device matches the test location in section 5.2.2, where no coverage was measured from C-IoT technologies.

When the device enters into a location where there is no C-IoT communication available, it performs a partial coverage communication sequence shown in figure 6.8 to try and maintain a fallback connection with the application server. The partial coverage communication sequence involves the following steps:

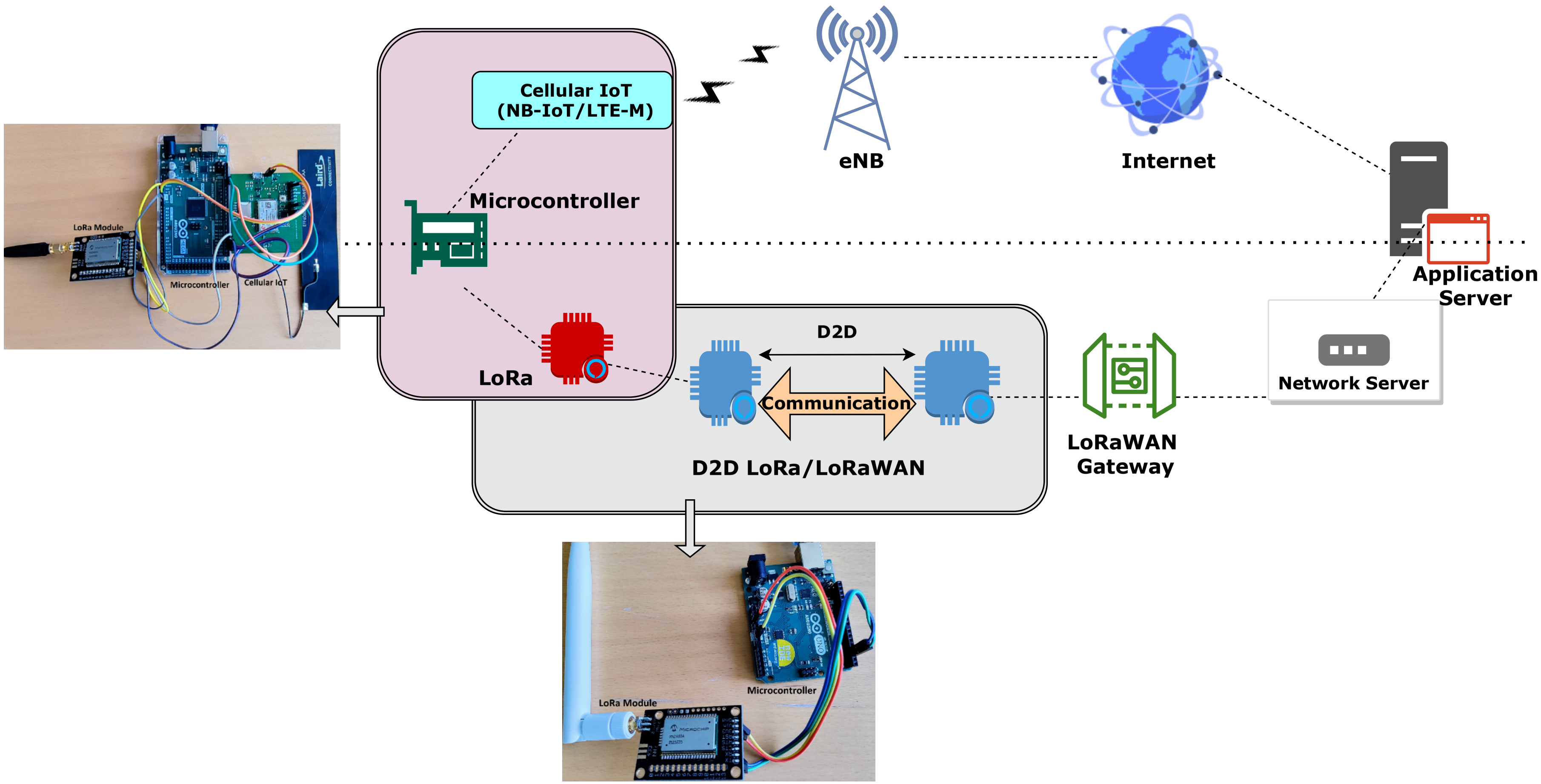


Figure 6.4: Device-TO-Device (D2D) Experimental Setup [122]

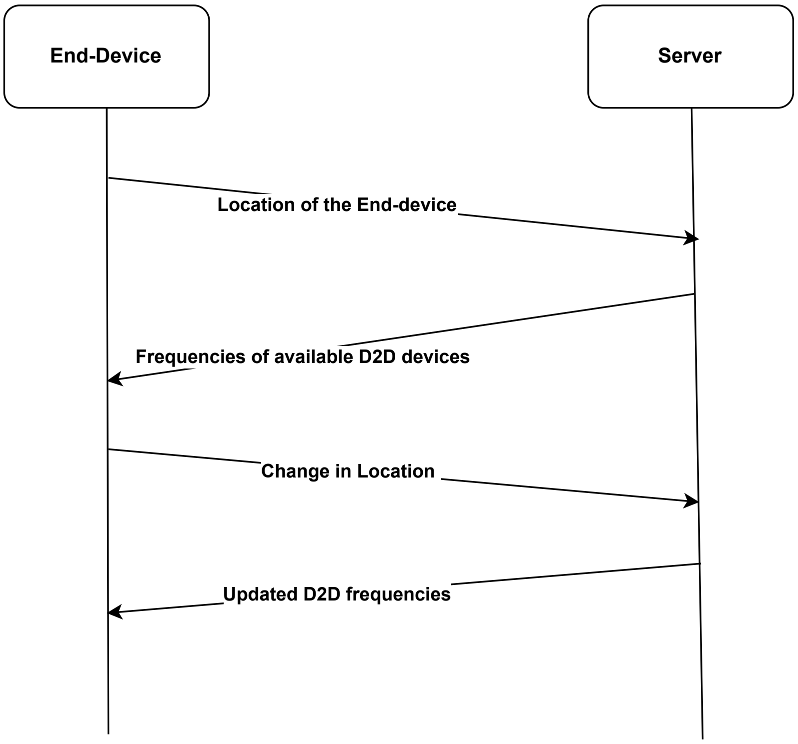


Figure 6.5: Initial D2D Setup Procedure [122]

Figure 6.6: Interaction between the end device and application server [122]

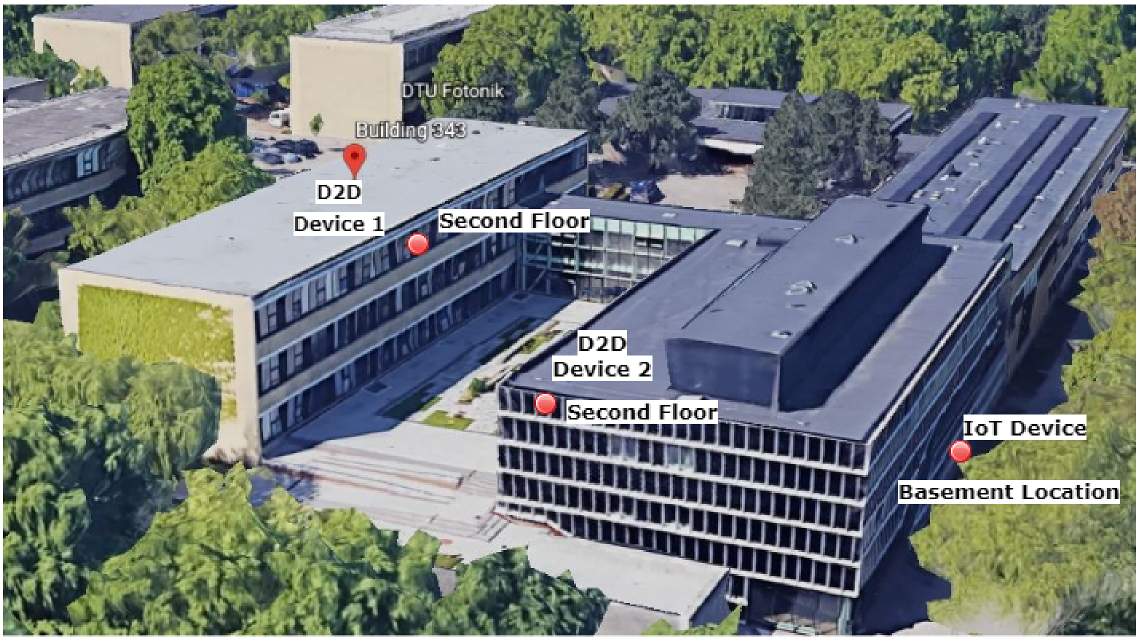


Figure 6.7: Deployment of the LoRa Devices [122]

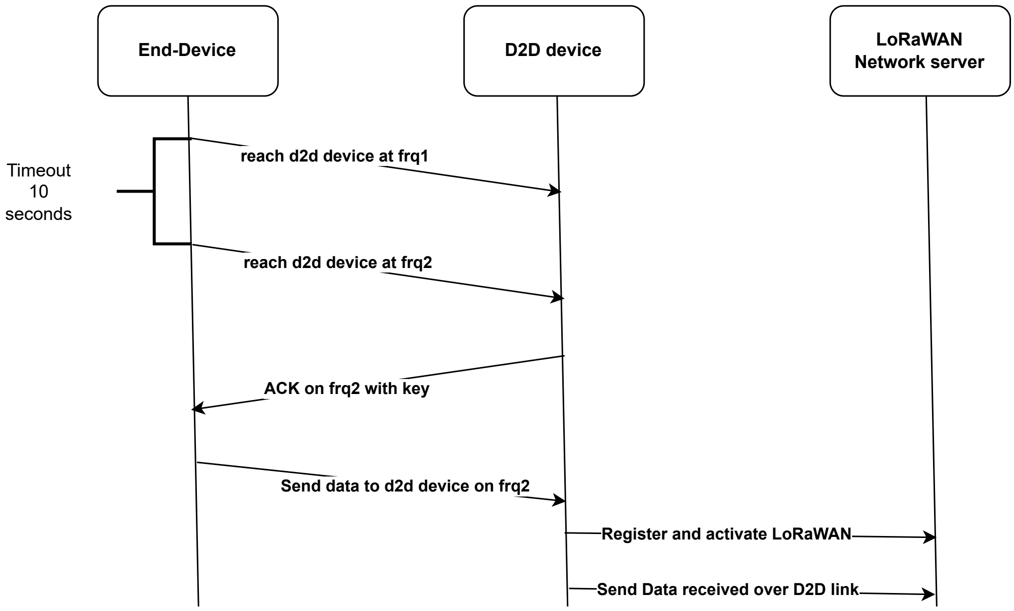


Figure 6.8: Partial Coverage Communication Sequence [122]

IoT device data at Network Server

Device EUI	Local time	Freq [MHz]	Data rate	RSSI (dBm)	SNR (dB)	FCntUp	Port	Payload
0004A30B00FE058E	2022-06-26 18:19:55 .482	867.9	SF12 BW125 4/5	-90	15	0	4	64 61 74 61 5f 64 31
0004A30B00FE058E	2022-06-26 17:52:22 .776	867.5	SF12 BW125 4/5	-95	1	0	4	64 61 74 61 5f 64 31
0004A30B00FE058E	2022-06-26 17:34:54 .928	867.7	SF12 BW125 4/5	-97	-22	0	4	64 61 74 61 5f 64 31

Figure 6.9: Data Delivery to LoRaWAN Network Server[122]

1. The device uses the list of frequencies stored in the "jasondata" file on the communication module to send a "Hello" message on every LoRa frequency. After the message is sent, the timeout period before the device switches to the next frequency is defined as 10 Sec. This is a pre-configured time setting and can be changed if required. This value was chosen by performing some initial Device-TO-Device (D2D) tests on the university campus and depended on the device's Spreading Factor (SF). During this experiment, the SF was set to 12. In LoRaWAN, the SP varies from SP7 to SP12; SP7 offers the highest bitrate, whereas SP12 offers the most extended range possible.

2. Once the LoRa Device-TO-Device (D2D) device has received the hello message, it sends back an acknowledgment and a secret key. This key is used by the IoT device to validate the identity of the LoRa Device-TO-Device (D2D) device.

3. Once the LoRa Device-TO-Device (D2D) secrete key is validated by the end device, it sends the actual payload to the LoRa Device-TO-Device (D2D) device using the same channel.

4. Once the LoRa Device-TO-Device (D2D) device receives the data, it changes its mode of operation from LoRa to LoRaWAN and starts the network registration process towards a LoRaWAN network server. Once the device is registered on the network server, it forwards the data received from the IoT end device to the network server. Depending upon the network server, the data can be pushed to our application server using REST APIs.

5. Once the data from the IoE end device has been forwarded using LoRaWAN, the Device-TO-Device (D2D) device changes its mode of operation again to LoRa.

Figure 6.9 shows the Device-TO-Device (D2D) messages sent from the IoT device using the Device-TO-Device (D2D) LoRa link. The data terminated in the LoRa network server is sent from the LoRa Device-TO-Device (D2D) devices. During the experiment, the data was sent thrice to the network server at three different time intervals. During the experiment, the last byte of the data was kept unique to find out which of the LoRa Device-TO-Device (D2D) device was sending the data. As it can be seen from the figure, the payload that was received by the LoRaWAN network server always had the same value 646174615f64**31**, which is a HEX representation of "data_d1" meaning for all three transmissions the Device-TO-Device (D2D) device one was used.

6.2 Summery

This chapter focuses on doing an experimental evaluation of different techniques to improve the reliability of the communication link between the end device and the remote server. The chapter explores possibilities of using Multi-RAT and D2D communication using LoRa to establish a fallback keep-alive channel between the end device and the server. The chapter also includes the development of a new prototype that combines both the approaches of Multi-RAT and D2D communication to improve the reliability of the C-IoT technologies when in outage scenario. Furthermore, the chapter includes the development and experimental validation of the LoRa Device-TO-Device (D2D) handshake method that can be used to transfer the end device data to the remote server by combining the use of LoRa and LoRaWAN technologies.

7 Conclusion and Future Works

The PhD thesis addresses the use of novel Cellular Internet of Things (C-IoT) technologies in telemedicine and continuous monitoring applications. The project evaluates a Mobile Network Operators (MNO) infrastructure's performance to support critical application needs. During the PhD project, several networks Key Performance Indicator (KPI) tests were conducted to evaluate the performance of C-IoT technologies. The primary question this PhD thesis focuses on is to verify the use of publicly deployed C-IoT networks for deploying critical Internet of Things (IoT) applications with demanding network Quality of Service (QoS) requirements. In order to make a realistic evaluation of the C-IoT networks, the author first builds a continuous ECG and HR monitoring application for patients with Cardiovascular Disease (CVD) by following Participatory Design methodology and then goes on to evaluate the C-IoT network Key Performance Indicator (KPI). The study is divided into two parts. Part one focuses on building an End-to-End (E2E) system, including the design of the end device and the data collection, storage, and visualization system, whereas the second part focuses on performance KPI evaluation of C-IoT networks.

The first part of the thesis focuses on designing different prototypes of the end device that can measure Electrocardiogram (ECG) and Heart Rate (HR) using Movesense and Savvy ECG monitors. The designed end device uses Long-Term Evolution Machine Type Communication (LTE-M) as the primary communication technology to send data to a data collection, storage, and visualization system called heartrater.live that is hosted on a server infrastructure at Technical University of Denmark (DTU). The part of the project follows Participatory Design that involves close collaboration with clinical researchers, medical professionals, and doctors from Denmark, Sweden, USA, and Japan.

The second part of the thesis focuses on performing several experiments to evaluate the KPI for C-IoT technologies. The results from testing C-IoT in an indoor and outdoor environment suggest that the technology can only support critical applications in some scenarios. When the C-IoT technologies are tested for continuous data transfer for several days, it was discovered that the KPI parameters differ between different MNO network deployments. Another outcome from the outdoor tests is that in the case of possible Evolved Node B (eNB) failure, the performance of Long-Term Evolution Machine Type Communication (LTE-M) suffers the most creating a complete outage scenario. In the case of the deep-indoor and remote-outdoor tests, the LTE-M network performance tends to suffer the most. When the C-IoT technologies were tested over three months, a noticeable performance difference was observed in the network KPI. Such variations in the network KPI can affect the performance of the IoT application. One way to adapt to the changing behavior of the C-IoT networks would be to deploy application-specific probes that are designed to periodically verify whether the KPI performance of C-IoT networks is within the expected range. This approach can help to adapt network degradation and help avoid complete loss of communication.

The testing of C-IoT networks in roaming scenarios exposes a significant performance gap between the NarrowBand-Internet of Things (NB-IoT) network deployed by the same MNO in different countries under similar network conditions. The possible reason for such a mismatched performance could be using different Radio Access Network (RAN) vendors and up to some extended mismatching radio parameter configurations. It was also discovered that different MNO had chosen different roaming architectures for their

C-IoT network deployment. Some MNO prefer sending data back to the home country of the Subscriber Identity Module (SIM), whereas others prefer pooling data to one country and sending it out through a common packet gateway. This approach might create challenges for the IoT applications that are part of the new directive on security of Network and Information systems (NIS) and the national autonomy restrictions imposed in the EU.

The network capacity tests indicate that in the case of LTE-M, only six simultaneous devices can send data without experiencing performance degradation when the cell is unloaded. If we assume that there are 4000 base stations in the whole of Denmark supporting LTE-M, we can only have a maximum of 72000 devices simultaneously sending data. This raises a big question on the scalability of the continuous remote heart motoring application designed during this project.

The power measurement tests conducted using different communication modules and different MNO networks highlight a big difference between these network's peak power consumption levels for different communication protocols in different coverage levels. The possible observed difference in the power levels could be because of the difference in the RAN vendors used for C-IoT deployments, different RAN parameters, different software releases of C-IoT, differences in the design of communication chipsets, etc.

Combining all the outcomes from different tests performed in this project, we believe that LTE-M and NB-IoT networks are not fully matured yet to support requirements imposed by continuous critical applications. Perhaps there is a need to create a substandard that can strictly impose the QoS requirements on these networks. Similar to Global System for Mobile Communications - Railway (GSM-R), which is a separate standard for railway communication over a generic GSM standard, there is a need to create NB-IoTc and LTE-Mc, where the c stands for explicitly designed for critical applications.

The contributions of the document can be summed up as follows:

Chapter 1 introduces the research questions and the current challenges in the field, and Chapter 2 focuses on providing core technical concepts and working principles of different technologies used during this PhD project.

Chapter 3 describes the different arguments for developing a custom IoT device that can communicate with multiple health sensors, receive data from them, and send it to a remote server location via C-IoT technologies. The discussion begins with the concept and the vision behind developing such a hardware device. The next part of the chapter focuses on developing three different prototypes of IoT devices, each with substantial improvements compared to the previous version. The chapter also includes evaluating different heart monitoring sensors and integrating those sensors with the IoT device. The final prototype closely resembles the vision for a continuous ECG and HR monitoring IoT device developed by following the Participatory Design methodology involving medical professionals, clinical researchers, and the CVD patients.

Chapter 4 evolution of the data collection and storage system and data visualization platform developed as a part of this PhD project. The different versions of the system prototype were developed closely following Participatory Design methodology involving medical professionals, clinical researchers, and CVD patients. The final version of the designed system can accept continuous monitoring data from the developed IoT prototype and once-a-day measurements containing step count, blood pressure, and sleep quality. The designed system features two visualization tools: a web portal called heartrater.live, designed specifically for medical professionals to visualize data from multiple professionals at once, and an Android app eMed, targeted for individual patients. The system can also process the incoming ECG data using an Machine Learning (ML) algorithm to classify the normal and abnormal patterns present in the data.

Chapter 5 focuses on evaluating the performance of C-IoT technologies in different physical environments. The tests were designed to measure E2E latency, bitrate, and Packet Drop (PD) in different network conditions. The tests were performed in indoor, deep indoor, outdoor, and remote outdoor environments. In most test scenarios, it was concluded that both LTE-M and NB-IoT perform well in good radio conditions. On the other hand, the network performance for LTE-M suffers the most in poor radio conditions than that of NB-IoT. When the C-IoT tests are performed after three months, the results indicate that the performance of the C-IoT varies over time.

Furthermore, similar tests were conducted in a roaming environment using a Nordic MNO. The test results show significant variation in the network performance, especially for NB-IoT. The observed differences might be due to the RAN deployed in these countries and the radio resources allocated for the C-IoT technologies. The chapter also gives an overview of power usage for different IoT modules using TCP and UDP to send data to an application server C-IoT technologies. The results highlight the difference between peak power consumption levels experienced by different protocols when data is sent using different communication modules using different MNO C-IoT deployments. The chapter includes experiments verifying the capacity of the eNB to handle simultaneous data transmission from several IoT devices. This test aimed to identify the scalability of the developed continuous remote ECG and HR monitoring application. The results indicate that in the case of LTE-M, one sector of the eNB can support a maximum of six continuously transmitting devices. In contrast, in the case of NB-IoT, that number is four. These results are MNO specific and may vary based on the available of LTE-M bandwidth and deployment modes NB-IoT (in-band, guard-band, etc.). Finally, the chapter includes tests conducted to understand the usability of the developed continuous ECG and HR monitoring system in other applications within healthcare. The results show that the LTE-M performance is dependent on the number of eNB deployed in the region, and it gets worse when the tests are conducted in areas where the eNB deployment is scattered.

Chapter 6 focuses on developing different solutions to improve the reliability of the data transfer by using Multiple Radio Access Technology (Multi-RAT) and Device-TO-Device (D2D) communication techniques using LoRa and LoRaWAN. The chapter experimentally validates the D2D handshake method that can send the end device data to a remote server by combining LoRa and LoRaWAN technologies to enable a ”keep alive” link in case the primary communication technology fails.

7.1 Future Research Directions

The continuous ECG and HR monitoring system can be used as an intervention for a Randomise Control Trial (RCT) in case of CVD patients rehabilitation study. Studying the likelihood of CVD patients and medical professionals adapting to using this system and including it in their everyday use will be exciting. Another area to focus on could be improving the range of health sensors that the device currently connects with; adding seamless integration with the weight scale, sleep sensor, pedometers, etc., will improve the quality of patient data collected and help remove human-induced errors. An obvious continuation of the developed system will include security features in the further development of the system. This can be done either by implementing low power, less complex

IoT encryption algorithms that will have less impact on the power usage of the devices.

Bibliography

[1] *IoT connections outlook*. https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/iot-connections-outlook. (Accessed on 07/19/2023).

[2] *Ericsson Mobility Visualizer*. https://www.ericsson.com/en/reports-and-papers/mobility-report/mobility-visualizer?f=1&ft=1&r=2,3,4,5,6,7,8,9&t=8&s=1,2,3&u=1&y=2020,2026&c=1. (Accessed on 07/19/2023).

[3] *WLR089U0*. https://www.microchip.com/en-us/product/WLR089U0. (Accessed on 07/20/2023).

[4] *Vision 2030: Low-Band Spectrum for 5G*. https://www.gsma.com/spectrum/wp-content/uploads/2022/07/5G-Low-Band-Spectrum-1.pdf/. (Accessed on 07/20/2023).

[5] *World Health Organization: Cardiovascular diseases*. https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1. (Accessed on 07/19/2023).

[6] B. Dinesen et al. "“Future patient” telerehabilitation for patients with heart failure Protocol for a randomized controlled trial". In: *Journal of Medical Internet Research* (2019), e14517. DOI: 10.2196/14517.

[7] Cathrine Skov Schacksen et al. "Effects of Telerehabilitation Interventions on Heart Failure Management (2015-2020)". In: *JMIR Rehabilitation Assistive Technologies* (2021), e29714. DOI: 10.2196/29714.

[8] Boback Ziaeian and Gregg C Fonarow. "Epidemiology and aetiology of heart failure." In: *Nature reviews. Cardiology, 13(6)* (2016), e29714. DOI: 10.1038/nrcardio.2016.25.

[9] L.J. Anderson and R.S. Taylor. "Cardiac rehabilitation for people with heart disease: An overview of Cochrane systematic reviews". In: *Elsevier Ireland Ltd* (2014), pp. 348–361. DOI: 10.1016/j.ijcard.2014.10.011.

[10] Josefine Dam Gade et al. "Predictors of Walking Activity in Patients With Systolic Heart Failure Equipped With a Step Counter: Randomized Controlled Trial". In: *Jmir Biomedical Engineering* (2020), e20776. DOI: 10.2196/20776.

[11] Stefanie Maria Werhahn et al. "Designing meaningful outcome parameters using mobile technology: a new mobile application for telemonitoring of patients with heart failure". In: *ESC Heart Failuer* (2019), pp. 516–525. DOI: 10.1002/ehf2.12425.

[12] Yuh Morimoto et al. "Web Portals for Patients with Chronic Diseases: Scoping Review of the Functional Features and Theoretical Frameworks of Telerehabilitation Platforms". In: *ESC Heart Failuer* (2022), e27759. DOI: 10.2196/27759.

[13] Jeremiah G. Zai Adrian H.and Ronquillo et al. "Assessing hospital readmission risk factors in heart failure patients enrolled in a telemonitoring program". In: *Hindawi Publishing Corporation* (2013), p. 305819. DOI: 10.1155/2013/305819.

[14] *Future Patient*. https://www.labwelfaretech.com/fp/. (Accessed on 07/19/2023).

[15] Astrid N.L. Hermans et al. "On-demand mobile health infrastructures to allow comprehensive remote atrial fibrillation and risk factor management through teleconsultation". In: *John Wiley and Sons Inc* (2020), pp. 1232–1239. DOI: 10.1002/clc.23469.

[16] Martin Manninger et al. "Role of wearable rhythm recordings in clinical decision making—The wEHRAbles project". In: *John Wiley and Sons Inc* (2020), pp. 1232–1239. DOI: 10.1002/clc.23404.

[17] Christopher Wilkinson and Antonella De Angeli. "Applying user centred and participatory design approaches to commercial product development". In: *Elsevier Ltd* (2014), pp. 614–631. DOI: 10.1016/j.destud.2014.06.001.
[18] Katrine Joensson et al. "Listening to the patients: using participatory design in the development of a cardiac telerehabilitation web portal". In: *Elsevier Ltd AME Publishing Company* (2019), p. 33. DOI: 10.21037/mhealth.2019.08.06.
[19] Ingrid C. Cnossen et al. "Listening to the patients: using participatory design in the development of a cardiac telerehabilitation web portal". In: *S. Karger AG* (2016), pp. 193–201. DOI: 10.1159/000441251.
[20] Susan Darlow and Kuang Yi Wen. "Development testing of mobile health interventions for cancer patient self-management: A review". In: *SAGE Publications Ltd* (2016), pp. 633–650. DOI: 10.1177/1460458215577994.
[21] *Cellular networks for Massive IoT*. https://www.ericsson.com/en/reports-and-papers/white-papers/cellular-networks-for-massive-iot--enabling-low-power-wide-area-applications. (Accessed on 07/20/2023).
[22] Michael A. Andersson et al. "Feasibility of Ambient RF Energy Harvesting for Self-Sustainable M2M Communications Using Transparent and Flexible Graphene Antennas". In: *Institute of Electrical and Electronics Engineers Inc.* (2016), pp. 5850–5857. DOI: 10.1109/ACCESS.2016.2604078.
[23] Kalpit Dilip Ballal. *Asset tracking using LP-WAN technologies*. Technical University of Denmark, 2019.
[24] *LoRa Alliance*. https://resources.lora-alliance.org/. (Accessed on 07/20/2023).
[25] Melvin P. Manuel and Kevin Daimi. "Implementing cryptography in LoRa based communication devices for unmanned ground vehicle applications". In: *Institute of Electrical and Electronics Engineers Inc.* (2021), p. 397. DOI: 10.1007/s42452-021-04377-y.
[26] P. Ramesh G.and Neha. *Design of LORA Based Tracking System for Military Personnel*. Springer Science and Business Media Deutschland GmbH, 2022. ISBN: 9811922802 , 9811922810 , 9789811922800 and 9789811922817.
[27] *LoRa: Symbol Generation*. http://www.sghoslya.com/p/lora-is-chirp-spread-spectrum.html. (Accessed on 07/20/2023).
[28] *LoRaWAN® Specification v1.0*. https://resources.lora-alliance.org/technical-specifications/lorawan-specification-v1-0. (Accessed on 07/20/2023).
[29] Jun Xu et al. "Narrowband internet of things: Evolutions, technologies, and open issues". In: *Institute of Electrical and Electronics Engineers Inc.* (2018), pp. 1449–1462. DOI: 10.1109/JIOT.2017.2783374.
[30] GSMA. *NB-IoT Deployment Guide to Basic Feature set Requirements*. https://www.gsma.com/iot/wp-content/uploads/2019/07/201906-GSMA-NB-IoT-Deployment-Guide-v3.pdf. (Accessed on 07/20/2023).
[31] J. Schlienz and D. Raddino. "Narrowband Internet of Things Whitepaper". In: *Rohde - Schwarz* (2018).
[32] Krzysztof Mateusz Malarski. *Reliable Architecture for Future Smart Communities*. Technical University of Denmark, 2021.
[33] Olof Liberg et al. *Cellular internet of things : Technologies, standards and performance*. Elsevier, 2017. ISBN: 012812458X , 012812458x , 0128124598 , 9780128124581 and 9780128124598. DOI: 10.1016/C2016-0-01868-5.
[34] Nitin Mangalvedhe, Rapeepat Ratasuk, and Amitava Ghosh. "NB-IoT deployment study for low power wide area cellular IoT". In: *Institute of Electrical and Electronics Engineers Inc.* (2016), p. 7794567. DOI: 10.1109/PIMRC.2016.7794567.

[35] Inc Qualcomm Technologies. *White Paper: Leading the LTE IoT evolution to connect the massive Internet of Things*. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/whitepaper_-_leading_the_lte_iot_evolution_to_connect_the_massive_internet_of_things_v1.0.pdf. (Accessed on 07/26/2023).
[36] *NB-IoT: a sustainable technology for connecting billions of devices*. https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/nb-iot-a-sustainable-technology-for-connecting-billions-of-devices. (Accessed on 07/20/2023).
[37] Rohde and Schwarz. *White Paper: Power saving methods for LTE-M and NB-IoT devices*. https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/premiumdownloads/premium_dl_brochures_and_datasheets/premium_dl_whitepaper/Power-saving-methods-for-LTE-M-and-NB-IoT-devices_wp_en_3609-3820-52_v0100.pdf. (Accessed on 07/20/2023).
[38] GSMA. *NB-IoT Deployment Guide to Basic Feature set Requirements*. https://www.gsma.com/newsroom/wp-content/uploads/CLP.28v1.0.pdf. (Accessed on 07/20/2023).
[39] Rohde and Schwarz. *White Paper: BE AHEAD IN CONNECTING EVERYTHING – LTE-M*. https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/premiumdownloads/premium_dl_brochures_and_datasheets/premium_dl_poster/Be_ahead_in_connecting_everything_LTE-M_po_en_5215-8087-82_v0200.pdf. (Accessed on 07/20/2023).
[40] Fernando Gregorio et al. *Signal processing techniques for power efficient wireless communication systems : practical approaches for RF impairments reduction*. Springer, 2020. ISBN: 3030324362 , 3030324370 , 9783030324360 and 9783030324377. DOI: 10.1007/978-3-030-32437-7_9.
[41] GSMA. *LTE-M Deployment Guide to Basic Feature Set Requirements*. https://www.gsma.com/iot/wp-content/uploads/2019/08/201906-GSMA-LTE-M-Deployment-Guide-v3.pdf. (Accessed on 07/20/2023).
[42] *The global fitness tracker market is projected to grow from* $36.34 billion in 2020 to$ *114.36 billion in 2028 at a CAGR of 15.4forecast period 2021-2028*. https://www.fortunebusinessinsights.com/fitness-tracker-market-103358. (Accessed on 07/21/2023).
[43] Erik Grönvall and Nervo Verdezoto. “Beyond self-monitoring: Understanding non-functional aspects of home-based healthcare technology : understanding non-functional aspects of home-based healthcare technology”. In: *ASSOC COMPUTING MACHINERY* (2013), pp. 587–596. DOI: 10.1145/2493432.2493495.
[44] S. Mirel et al. “The Use of Medical Devices in Self Monitoring of Chronic Diseases”. In: *Springer Verlag* (2011), pp. 48–53. DOI: 10.1007/978-3-642-22586-4_12.
[45] Stinne Aaløkke Ballegaard, Thomas Riisgaard Hansen, and Morten Kyng. “Healthcare in Everyday Life - Designing Healthcare Services for Daily Life”. In: *Association for Computing Machinery* (2008), pp. 1807–1816. DOI: 10.1145/1357054.1357336.
[46] Erik Grönvall and Morten Kyng. “On participatory design of home-based healthcare”. In: *SPRINGER LONDON LTD* (2013), pp. 389–401. DOI: 10.1007/s10111-012-0226-7.
[47] Rikke Aarhus, Stinne Aaløkke Ballegaard, and Thomas Riisgaard Hansen. “Bridging home and hospital through healthcare technology”. In: *Kluwer Academic Publishers* (2009), pp. 63–82. DOI: 10.1007/978-1-84882-854-4_4.
[48] *AliveCor: Hoempage*. https://www.alivecor.com/. (Accessed on 07/21/2023).
[49] *Cortrium: Homepage*. https://www.cortrium.com. (Accessed on 07/21/2023).

[50] *Karlia:Homepage*. https://store.kardia.com/products/kardiamobile. (Accessed on 07/21/2023).

[51] *SparkFun: Device overview*. https://www.sparkfun.com/products/12650. (Accessed on 07/21/2023).

[52] *Movesense: Homepage*. https://www.movesense.com. (Accessed on 07/21/2023).

[53] *Savvy: Homepage*. http://www.savvy.si. (Accessed on 07/21/2023).

[54] *Movesense: productpage*. https://www.movesense.com/product/movesense-medical/. (Accessed on 07/21/2023).

[55] *Espressif: Homepage*. https://www.espressif.com/en/products/devkits/esp32-devkitc. (Accessed on 07/21/2023).

[56] *Espressif: Documentation*. https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html. (Accessed on 07/21/2023).

[57] Kalpit Dilip Ballal, Radheshyam Singh, and Lars Dittmann. "Real-Time Remote Heart Monitoring Using Cellular IoT Network". In: 2022.

[58] Saqib Hameed and Oliver Isling Pærregaard. *Remote monitoring of heart patients using 5G & IoT*. DTU Department of Photonics Engineering, 2021.

[59] *InfluxDB: Homepage*. https://www.influxdata.com/. (Accessed on 07/21/2023).

[60] *ATSAMD21G18: Product Page*. https://www.microchip.com/en-us/product/ATsamd21g18l. (Accessed on 07/21/2023).

[61] *NINA-B31 series: Product Page*. https://www.u-blox.com/en/product/nina-b31-series-u-connect. (Accssed on 07/21/2023).

[62] *STM TESEO-LIV3R: Product Page*. https://www.st.com/en/positioning/teseo-liv3r.html#overview. (Accessed on 07/21/2023).

[63] *STM TESEO-LIV3R:buy page*. https://www.mouser.dk/ProductDetail/STMicroelectronics/TESEO-LIV3R. (Accessed on 07/21/2023).

[64] *SARA-R4 series: Product Page*. https://www.u-blox.com/en/product/sara-r4-series. (Accessed on 07/21/2023).

[65] Mohammad Nasar and Mohammad Abu Kausar. "Suitability of influxdb database for iot applications". In: *Blue Eyes Intelligence Engineering and Sciences Publication* (2013), pp. 1850–1857. DOI: 10.35940/ijitee.J9225.0881019.

[66] *Grafana: Homepage*. https://grafana.com. (Accessed on 07/21/2023).

[67] *PostgreSQL: Homepage*. https://www.postgresql.org/. (Accessed on 07/21/2023).

[68] Mathilde Tannebæk and Naja Jean Larsen. *Development of API and User Interface for visualization of Heart sensor data*. DTU Department of Applied Mathematics and Computer Science, 2021.

[69] *Axios: Homepage*. https://axios-http.com/docs/intro. (Accessed on 07/21/2023).

[70] *Node.js: Homepage*. https://nodejs.org/en. (Accessed on 07/21/2023).

[71] Mohamed Sraitih, Younes Jabrane, and Amir Hajjam El Hassani. "An automated system for ECG arrhythmia detection using machine learning techniques". In: *MDPI* (2021), pp. 1850–1857. DOI: 10.3390/jcm10225450.

[72] Ali Haider Khan, Muzammil Hussain, and Muhammad Kamran Malik. "Arrhythmia Classification Techniques Using Deep Neural Network". In: *MDPI* (2021), p. 9919588. DOI: 10.1155/2021/9919588.

[73] Chia Yen Yang, Pin Chen Chen, and Wen Chen Huang. "Cross-Domain Transfer of EEG to EEG or ECG Learning for CNN Classification Models". In: *MDPI* (2023), p. 9919588. DOI: 10.3390/s23052458.

[74] Malak Fora, Sajidah Al-Hammouri, and Awad Al-Zaben. "ECG Abnormality Detection from PPG Signal". In: *Institute of Electrical and Electronics Engineers Inc.* (2019), pp. 103–106. DOI: 10.1109/JEEIT.2019.8717512.

[75] Elena Merdjanovska and Aleksandra Rashkovska. “Patient-Specific Heartbeat Classification in Single-Lead ECG using Convolutional Neural Network”. In: *Institute of Electrical and Electronics Engineers Inc.* (2021), pp. 932–936. DOI: 10.1109/EMBC46164.2021.9630366.
[76] Deena P. Francis et al. “Classification of colorimetric sensor data using time series”. In: *SPIE - International Society for Optical Engineering* (2021), p. 118700L. DOI: 10.1117/12.2600182.
[77] Jasmeet Kaur and Anil Kumar. *Speech Emotion Recognition Using CNN, k-NN, MLP and Random Forest*. Springer Science and Business Media Deutschland GmbH, 2021. ISBN: 9811596468 , 9811596476 , 9789811596469 and 9789811596476. DOI: 10.1007/978-981-15-9647-6_39.
[78] Yun Ma. *The application of machine learning in the field of intelligent medical data analysis*. DTU Department of Electrical and Photonics Engineering, 2022.
[79] *MIT-BIH Database Distribution*. https://ecg.mit.edu/. (Accessed on 07/22/2023).
[80] *PTB-XL, a large publicly available electrocardiography dataset*. https://physionet.org/content/ptb-xl/1.0.3/. (Accessed on 07/22/2023).
[81] Radheshyam Singh and Ying Yan. “Empirical Test for Coverage Evaluation of NB-IoT Network”. In: *IEEE* (2021), p. 6. DOI: 10.1109/ICAC353642.2021.9697216.
[82] Samir Dawaliby, Abbas Bradai, and Yannis Pousset. “In depth performance evaluation of LTE-M for M2M communications”. In: *IEEE Computer Society* (2016). DOI: 10.1109/WiMOB.2016.7763264.
[83] Hassan Malik et al. “NB-IoT Network Field Trial: Indoor, Outdoor and Underground Coverage Campaign”. In: *Institute of Electrical and Electronics Engineers Inc.* (2019), pp. 537–542. DOI: 10.1109/IWCMC.2019.8766568.
[84] Subho Shankar Basu et al. “Experimental Performance Evaluation of NB-IoT”. In: *IEEE Computer Society* (2019), p. 8923221. DOI: 10.1109/WiMOB.2019.8923221.
[85] Massimo Ballerini et al. “NB-IoT Versus LoRaWAN: An Experimental Evaluation for Industrial Applications”. In: *IEEE Computer Society* (2020), pp. 7802–7811. DOI: 10.1109/TII.2020.2987423.
[86] Ahmed Mahmood and Saima Zafar. “Performance analysis of narrowband internet of things (nb-iot) deployment modes”. In: *Institute of Electrical and Electronics Engineers Inc.* (2019), p. 9022748. DOI: 10.1109/INMIC48123.2019.9022748.
[87] Krzysztof Mateusz Malarski et al. “Demonstration of NB-IoT for Maritime Use Cases”. In: *Institute of Electrical and Electronics Engineers Inc.* (2018), pp. 106–108. DOI: 10.1109/NOF.2018.8598067.
[88] Jubin E. Sebastian et al. “Test and Measurement of LPWAN and Cellular IoT Networks in a Unified Testbed”. In: *Institute of Electrical and Electronics Engineers Inc.* (2019), pp. 1521–1527. DOI: 10.1109/INDIN41052.2019.8972256.
[89] Andre Sorensen et al. “Modeling and Experimental Validation for Battery Lifetime Estimation in NB-IoT and LTE-M”. In: *Institute of Electrical and Electronics Engineers Inc.* (2022), pp. 9804–9819. DOI: 10.1109/JIOT.2022.3152173.
[90] Sebastian E. Jubin and Axel1 Sikora. “Performance Measurements of Narrowband-IoT Network in Emulated and Field Testbeds”. In: *IEEE* (2019), pp. 780–5. DOI: 10.1109/IDAACS.2019.8924357.
[91] *PYCOM FIPY: Product page*. https://docs.pycom.io/datasheets/development/fipy/. (Accessed on 07/25/2023).
[92] *Rude Skov*. https://naturstyrelsen.dk/naturoplevelser/naturguider/rude-skov/. (Accessed on 07/26/2023).

[93] Kalpit Dilip Ballal, Lars Dittmann, and Sarah Ruepp. "Experimental Performance Evaluation of Cellular IoT networks in Deep-Indoor Scenario". In: 2021, p. 6. DOI: 10.23919/PEMWN53042.2021.9664690.
[94] *Strava: Rude Skove*. √. (Accessed on 07/30/2023).
[95] Kalpit Dilip Ballal, Lars Dittmann, and Sarah Ruepp. "Experimental Evaluation of Roaming Performance of Cellular IoT Networks". In: 2022, pp. 386–391. DOI: 10.1109/ICUFN55119.2022.9829590.
[96] Dino Flore. *3GPP Standards for the Internet-of-Things*. https://www.3gpp.org/images/presentations/3GPP_Standards_for_IoT.pdf. (Accessed on 07/25/2023).
[97] 3GPP. *Technical Report: Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT) (Release 13)*. https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit-2. (Accessed on 07/25/2023).
[98] Hassan Malik et al. "NarrowBand-IoT Performance Analysis for Healthcare Applications". In: *Elsevier B.V.* (2018), pp. 1077–1083. DOI: 10.1016/j.procs.2018.04.156.
[99] Mads Lauridsen et al. "Coverage and capacity analysis of LTE-M and NB-IoT in a rural area". In: *Institute of Electrical and Electronics Engineers Inc.* (2016), p. 7880946. DOI: 10.1109/VTCFall.2016.7880946.
[100] Tin Yu Wu et al. "Persistent Periodic Uplink Scheduling Algorithm for Massive NB-IoT Devices". In: *MDPI* (2018), p. 2875. DOI: 10.3390/s22082875.
[101] Haider A.H. Alobaidy et al. "Real-World Evaluation of Power Consumption and Performance of NB-IoT in Malaysia". In: *Institute of Electrical and Electronics Engineers Inc.* 9 (2022), pp. 11614–11632. DOI: 10.1109/JIOT.2021.3131160.
[102] Mads Lauridsen et al. "An Empirical NB-IoT Power Consumption Model for Battery Lifetime Estimation". In: *Institute of Electrical and Electronics Engineers Inc.* (2018), pp. 1–5. DOI: 10.1109/VTCSpring.2018.8417653.
[103] Pascal Jorke, Robert Falkenberg, and Christian Wietfeld. "Power Consumption Analysis of NB-IoT and eMTC in Challenging Smart City Environments". In: *Institute of Electrical and Electronics Engineers Inc.* (2019), p. 8644481. DOI: 10.1109/GLOCOMW.2018.8644481.
[104] Rudi Hernowo et al. "Power Consumption Optimization for Flood Monitoring System using NB-IoT". In: *Institute of Electrical and Electronics Engineers Inc.* (2022), pp. 58–63. DOI: 10.1109/ISRITI56927.2022.10052891.
[105] Chunhe Song et al. "Power Consumption and Coverage Test of NB-IoT in the Substation Edge Computing Environment". In: *Springer Verlag* (2019), pp. 419–429. DOI: 10.1007/978-3-030-24265-7_36.
[106] Svetoslav Duhovnikov et al. "Power Consumption Analysis of NB-IoT Technology for Low-Power Aircraft Applications". In: *Institute of Electrical and Electronics Engineers Inc.* (2019), pp. 719–723. DOI: 10.1109/WF-IoT.2019.8767234.
[107] *Pycom GPY: Product Page*. https://docs.pycom.io/datasheets/development/gpy/. (Accessed on 07/30/2023).
[108] Dino Flore. *nRF Power Profiler Kit II: product page*. https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit-2. (Accessed on 07/25/2023).
[109] *Cleveland Clinic: Thrombosis*. https://my.clevelandclinic.org/health/diseases/22242-thrombosis. (Accessed on 07/30/2023).
[110] *TSMA6B: Product Page*. https://www.rohde-schwarz.com/cz/products/test-and-measurement/network-data-collection/rs-tsmx-drive-and-walk-test-scanner_63493-526400.html. (Accessed on 07/25/2023).

[111] Jason Godfrey. *Improve coverage with automatic network selection*. https://onomondo.com/blog/improve-coverage-with-automatic-network-selection/. (Accessed on 07/26/2023).

[112] *ETSI: TETRA Introduction*. https://www.etsi.org/technologies/tetra. (Accessed on 07/28/2023).

[113] J Schlienz and A Roessler. *Device to Device Communication in LTE Whitepaper*. https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1ma264/1MA264_0e_D2DComm.pdf. (Accessed on 07/25/2023).

[114] 3GPP. *Study on further enhancements to LTE Device to Device (D2D), User Equipment (UE) to network relays for Internet of Things (IoT) and wearables*. Technical Report (TR) 36.746. 3rd Generation Partnership Project (3GPP), 2018.

[115] Krzysztof Mateusz Malarski, Kalpit Dilip Ballal, and Sarah Ruepp. "D2D-enabled Failure-tolerance in Cellular IoT". In: *IEEE* (2021), p. 5. DOI: 10.1109/NoF52522.2021.9609924.

[116] Konstantin Mikhaylov et al. "D2D communications in LoRaWAN Low Power Wide Area Network: From idea to empirical validation". In: *Institute of Electrical and Electronics Engineers Inc.* (2017), pp. 737–742. DOI: 10.1109/ICCW.2017.7962746.

[117] Akram H. Jebril et al. "Overcoming limitations of LoRa physical layer in image transmission". In: *MDPI AG* (2018), p. 3257. DOI: 10.3390/s18103257.

[118] Jaehyu Kim and Jooseok Song. "A Secure Device-To-Device Link Establishment Scheme for LoRaWAN". In: *Institute of Electrical and Electronics Engineers Inc.* (2018), pp. 2153–2160. DOI: 10.1109/JSEN.2017.2789121.

[119] Nazmus Saqib et al. "D2D-LoRa Latency Analysis: An Indoor Application Perspective". In: *Institute of Electrical and Electronics Engineers Inc.* (2021), pp. 29–34. DOI: 10.1109/WF-IoT51360.2021.9595324.

[120] Luca Sciullo et al. "LOCATE: A LoRa-based mObile emergenCy mAnagement sysTEm". In: *Institute of Electrical and Electronics Engineers Inc.* (2018), p. 8647177. DOI: 10.1109/GLOCOM.2018.8647177.

[121] Luis Henrique De Oliveira Alves et al. "Network-Coded Cooperative LoRa Network With D2D Communication". In: *Institute of Electrical and Electronics Engineers Inc.* (2022), pp. 4997–5008. DOI: 10.1109/JIOT.2021.3107784.

[122] Kalpit Dilip Ballal et al. "Implementation of Failure Recovery in Emergency IoT Applications Using D2D in LoRa/LoRaWAN". In: 2022, pp. 41–46. DOI: 10.1109/CAMAD55695.2022.9966914.

[123] Kalpit Dilip Ballal et al. "Reliable Internet of Things for Health Care Technologies". In: 2023. DOI: 10.2196/41191.

[124] Kalpit Dilip Ballal, Lars Dittmann, and Birthe Dinesen. "Remote Monitoring Heart Failure patients using C-IoT technologies". In: 2023.

[125] Kalpit D. Ballal et al. "IoT Devices Reliability Study: Multi-RAT Communication". In: 2020, p. 9221163. DOI: 10.1109/WF-IoT48130.2020.9221163.

[126] Radheshyam Singh et al. "Overview of Drone Communication Requirements in 5G". In: 2023, pp. 3–16. DOI: 10.1007/978-3-031-20936-9_1.

[127] Radheshyam Singh et al. "Assessment of Cellular Coverage for a Smart Ambulance Use Case". In: 2022.

[128] Krzysztof Mateusz Malarski et al. "Internet of Reliable Things: Toward D2D-enabled NB-IoT". In: 2020, pp. 196–201.

[129] Radheshyam Sing et al. "Empirical Test for Coverage Evaluation of NB-IoT Network". In: 2021, p. 6. DOI: 10.1109/ICAC353642.2021.9697216.

[130] Radheshyam Sing et al. "An investigation of 5G, LTE, LTE-M and NB-IoT coverage for drone communication above 450 feet". In: 2023, p. 6.

[131] *ANT-GSMMQB: Performance Data*. https://www.rfsolutions.co.uk/downloads/6287ac749fbb4108ANT-GSMMQB-7.pdf. (Accessed on 07/28/2023).
[132] *LTE frequency bands*. https://en.wikipedia.org/wiki/LTE_frequency_bands. (Accessed on 07/28/2023).
[133] *Electrocardiography*. https://en.wikipedia.org/wiki/Electrocardiography. (Accessed on 07/26/2023).
[134] *Atrial Fibrillation*. https://medschool.co/tests/ecg-tachyarrhythmias/atrial-fibrillation. (Accessed on 07/28/2023).

List of publications

1. Kalpit Dilip Ballal, Lars Dittmann, and Sarah Ruepp. “Experimental Performance Evaluation of Cellular IoT networks in Deep-Indoor Scenario”. In: 2021, p. 6. DOI: 10.23919/PEMWN53042.2021.9664690
2. Kalpit Dilip Ballal, Lars Dittmann, and Sarah Ruepp. “Experimental Evaluation of Roaming Performance of Cellular IoT Networks”. In: 2022, pp. 386–391. DOI: 10.1109/ICUFN55119.2022.9829590
3. Kalpit Dilip Ballal, Radheshyam Singh, Stanley Chukwuebuka Nwabuona, Lars Dittmann, and Sarah Ruepp. “Implementation of Failure Recovery in Emergency IoT Applications Using D2D in LoRa/LoRaWAN”. in: 2022, pp. 41–46. DOI: 10.1109/CAMAD55695.2022.9966914
4. Kalpit Dilip Ballal, Lars Dittmann, Sarah Ruepp, and Birthe Dinesen. “Reliable Internet of Things for Health Care Technologies”. In: 2023. DOI: 10.2196/41191
5. Kalpit Dilip Ballal, Lars Dittmann, and Birthe Dinesen. “Remote Monitoring Heart Failure patients using C-IoT technologies”. In: 2023
6. Kalpit D. Ballal, Lars DIttmann, Sarah Ruepp, and Martin Nordal Petersen. “IoT Devices Reliability Study: Multi-RAT Communication”. In: 2020, p. 9221163. DOI: 10.1109/WF-IoT48130.2020.9221163
7. Kalpit Dilip Ballal, Radheshyam Singh, and Lars Dittmann. “Real-Time Remote Heart Monitoring Using Cellular IoT Network”. In: 2022

Other articles published during the PhD

1. Radheshyam Singh, Kalpit Dilip Ballal, Michael Stübert Berger, and Lars Dittmann. “Overview of Drone Communication Requirements in 5G”. in: 2023, pp. 3–16. DOI: 10.1007/978-3-031-20936-9_1
2. Radheshyam Singh, Kalpit Dilip Ballal, Stanley Chukwuebuka, Berger S. Michael, Lars Dittmann, and Sarah Ruepp. “Assessment of Cellular Coverage for a Smart Ambulance Use Case”. In: 2022
3. Krzysztof Mateusz Malarski, Farnaz Moradi, Kalpit Dilip Ballal, Lars Dittmann, and Sarah Ruepp. “Internet of Reliable Things: Toward D2D-enabled NB-IoT”. in: 2020, pp. 196–201
4. Krzysztof Mateusz Malarski, Kalpit Dilip Ballal, and Sarah Ruepp. “D2D-enabled Failure-tolerance in Cellular IoT”. in: *IEEE* (2021), p. 5. DOI: 10.1109/NoF52522.2021.9609924
5. Radheshyam Sing, Ying Yan, Kalpit Dilip Ballal, and Lars Dittmann. “Empirical Test for Coverage Evaluation of NB-IoT Network”. In: 2021, p. 6. DOI: 10.1109/ICAC353642.2021.9697216

Submitted papers

Radheshyam Sing, Jes H. Jepsenb, Kalpit Dilip Ballal, Stanley Chukwuebuka, Michael Bergera, and Dittmann Lars. “An investigation of 5G, LTE, LTE-M and NB-IoT coverage for drone communication above 450 feet”. In: 2023, p. 6

Thesis Collaborations

1. Saqib Hameed and Oliver Isling Pærregaard. *Remote monitoring of heart patients using 5G & IoT*. DTU Department of Photonics Engineering, 2021

   The collaboration with students during the time of the thesis focused on the following aspects:

   (a) Improving the stability of the Bluetooth Low Energy (BLE) connection link.

   (b) Validation of the working of the final prototype.

   (c) Validation of the working of End-to-End (E2E) continuous Electrocardiogram (ECG) and Heart Rate (HR) monitoring system in different environments.

   (d) Development of the Android app eMed.

2. Mathilde Tannebæk and Naja Jean Larsen. *Development of API and User Interface for visualization of Heart sensor data*. DTU Department of Applied Mathematics and Computer Science, 2021

   The collaboration with students during the time of the thesis focused on the following aspects:

   (a) Creating a relevant data visualization layer to view the monitoring data sent by patients. This included data points measured once a day and the possibility of viewing the ECG and HR measurements in realtime.

   (b) A generic approach to creating different components in the system which can function in decoupled state and can easily be extended or replaced if required.

   (c) Ability to retrieve data from the backend system by means of Representational State Transfer (REST) Application Programming Interface (API) that can be consumed by other applications.

   (d) Validation of the designed system by experts in the User Experience field and medical professionals.

   This section 4.2.2, describes the key outcomes from the bachelor project and highlights some important changes in the system overview.

3. Yun Ma. *The application of machine learning in the field of intelligent medical data analysis*. DTU Department of Electrical and Photonics Engineering, 2022

   The collaboration with students during the time of the thesis focused on the following aspects:

   (a) Design and verification of Machine Learning (ML) model to classify the incoming ECG into normal and abnormal pattern.

A Clinical Feedback

A.1 Workshop with Patients

Aim:

The aim of the workshop was to understand requirements from patients and their choices when it comes to the choices for ECG sensors and HR monitors.

Date:

June 2019

Participants:

CVD patients, clinical researchers, medical doctors from Viborg and Skive cardiac rehabilitation center.

Place:

Viborg, Denmark.

Outcome related to the project:

Information and requirements about the expected behaviour of the continuous remote monitoring system according to the end users (i.e., medical professionals and CVD patients)

A.2 Danish Japanese Technology Workshop

Aim:

The aim of the telecardiology workshop was to discuss use of different technologies that are used in treating patients with CVD

Date:

June 2019

Participants:

clinical researchers in the field of telemedicine and telecardiology, industries from Japan designing healthcare equipment and sensors, technical experts in the field of telemedicine.

Place:

Aalborg University, Denmark.

Outcome related to the project:

Understanding the Sate-of-Art when it come to healthcare devices used in telecardiology, digital platforms platform, and conceptual validations for a need of continuous monitoring using of IoT technologies.

A.3 TTRN PhD Summer School: 01

Aim:

The aim of the summer school was to present the conceptual idea of remote monitoring system using IoT technologies and receive feedback.

Date:

August 2019

Participants:

Telemedicine and telemonitoring researchers from USA, AAL, SDU, and DTU.

Place:

SDU University, Denmark.

Outcome related to the project:

Basic understanding of the clinical domain and what not to do when designing a critical monitoring system

A.4 Project Workshop: 01

Aim:

The aim of the workshop was to kick start the discuss for using Low Power Wide Area Network (LPWAN) IoT technologies for remote monitoring of CVD paients.

Date:

September 2019

Participants:

Telemedicine researchers from AAU and technical experts.

Place:

Aalborg University, Denmark.

Outcome related to the project:

1. Continuous vs on-demand use of remote monitoring of CVD patients using LPWAN IoT.
2. Different use cases and possible scenarios, as well as stakeholders that can benifit from the use of continuous remote heart monitoring IoT system.

A.5 Outpatient visit

Aim:

The aim of the activity was to speak with actual CVD patients to understand their feedback on use of sensors and IoT to collect continuous realtime data remotely and allowing them to reduce the hospital visits. The patients were also asked to give their feedback on the device prototype and the sensor.

Date:

January 2020

Participants:

Telemedicine researchers from AAU and CVD patients.

Place:

Skive, Denmark.

Outcome related to the project:

1. Recommended changes in the design of the prototype device.
2. Possible locations and places where they would like to wear the device and use such remote monitoring system.

A.6 Outpatient visit

Aim:

The aim of the activity was to speak with actual CVD patients to understand their feedback on use of sensors and IoT to collect continuous realtime data remotely and allowing them to reduce the hospital visits. The patients were also asked to give their feedback on the device prototype and the sensor.

Date:

January 2020

Participants:

Telemedicine researchers from AAU and CVD patients.

Place:

Skive, Denmark.

Outcome related to the project:

1. Recommended changes in the design of the prototype device.
2. Possible locations and places where they would like to wear the device and use such remote monitoring system.

A.7 Project Workshop: 02

Aim:

The aim was the workshop was to understand the clinical requirements from the data collection device as well as data visualisation platform from medical doctors, cardiac nurses, researchers from AAU.

Date:

February 2020

Participants:

Telemedicine researchers from AAU, cardic nurses, medical doctors, clinical technician.

Place:

Skive, Denmark.

Outcome related to the project:

1. feedback on current challenges while treading patients with CVD.
2. use of technology and CVD patients expectations.
3. Recommended changes in the design of the prototype device.

A.8 TTRN PhD Summer School: 02

Aim:

The aim of the summer school was to present the realtime HR monitoring device prototype and receive feedback from domain experts.

Date:

August 2020

Participants:

Telemedicine and telemonitoring researchers from USA, AAL, SDU, and DTU.

Place:

Virtual Summer School.

Outcome related to the project:

General feedback on use of prototype design and the use of HR sensor. Feedback on the design changes to the end device and recommended changes to ML model.

A.9 Project Workshop: 03

Aim:

The aim of the workshop was to possibilities of continuous ECG and HR using sensors like Cotrium, Movesense, etc. to send data to a remote server.

Date:

September 2020

Participants:

Telemedicine researchers from AAU, clinical technician.

Place:

Aalborg University, Denmark.

Outcome related to the project:

- Possibilities of outpatient visits
- To build a protype using ECG and HR sensors and send data to a remote server for storage and visualisation.

A.10 TTRN PhD Summer School: 03

Aim:

The aim of the summer school was to present the realtime ECG and HR monitoring device prototype coupled with the live demo of the heartrate.live visualisation sytem.

Date:

August 2021

Participants:

Telemedicine and telemonitoring researchers from USA, AAL, SDU, and DTU.

Place:

Virtual Summer School.

Outcome related to the project:

General feedback on use of new E2E ECG and HR monitoring system.

A.11 Workshop: Developing a Real-time remote monitoring system for patients with Atrial Fibrillation

Aim:

- To demonstrate a real-time remote monitoring system for patients with Atrial Fibrillation
- To get feedback from specialist in atrial fibrillation on requirements to the system for future remote monitoring of patients with Atrial Fibrillation

Date:

November 2021

Participants:

Egon Toft, Research Director, Forskning, Uddannelse og Innovation, clinical researchers from AAU.

Place:

Aalborg University, Denmark

Outcome related to the project:

Overall feedback was very positive with minor suggested changes in the outer design of the device and the visualisation layers. The next suggested next steps were to use the system in a clinical study.

A.12 Project Workshop: 03

Aim:

- To demonstrate a real-time remote monitoring system for patients with Atrial Fibrillation
- To get feedback from specialist in atrial fibrillation on requirements to the system for future remote monitoring of patients with Atrial Fibrillation

Date:

May 2022

Participants:

Clinical researchers and technical domain experts from AAU.

Place:

Aalborg University, Denmark

Outcome related to the project:

The overall feedback for the system was positive. The specific feedback for each part of the system is as follows:

- **end device:** It was suggested to design a waist belt clip for housing the end device PCB. To make the enclosure water-resistant, and add other ways of charging the device. Another feedback was to include display onto the device that can display local recording of the ECG and HR.
- **Data collection, storage, and visualisation system:** The visualisation part of the system was very simple to navigate and understand. To add a ECG pause button, so that people looking signal can examine the signal without worrying about loosing the data.

A.13 Project Workshop: 05

Aim:

- To demonstrate a real-time remote monitoring system for patients with Atrial Fibrillation
- TO identify other areas withing telemedicine and telemonitoring where these devices could be used.

Date:

June 2022

Participants:

Troels Wienecke, Department of Neurology, Zealand University Hospital, Roskilde, Denmark

Place:

Roskilde Hospital, Denmark

Outcome related to the project:

The overall feedback for the system was positive and indicated that the designed system can be useful in detecting early signs of atrial fibrillation after suffering from acute stroke.

A.14 UX design Questionnaires and eMed app evaluation

The UX design questionnaires were part of the BSc Thesis collaboration [68].

A.14.1 First UX Questionnaire

Early user experience testing

11 svar

Offentliggør analyse

Prototype

Hvad var dit første indtryk af applikationen?

9 svar

wowsa

Dejlig nem og overskuelig

Behageligt farvevalg

Flot og informationsrigt. Der er meget data.

Super overskueligt og nemt at aflæse

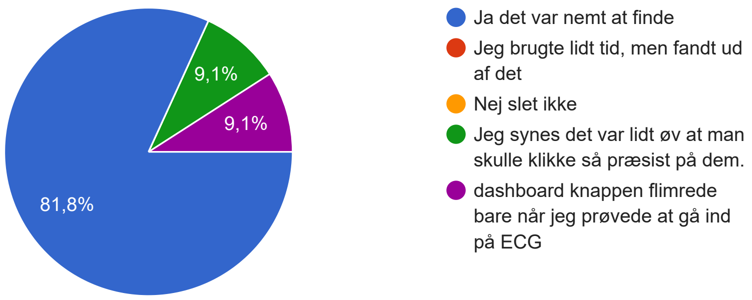
Den var simpel of ligetil. Akse tick-labels i ECG live virker forvirrende når de er så lange. Måske ville det bære bedre bare at have en akse med tid i [s] og så ikke angive det præcise klokkeslæt.

It looks very nice

Virkelig flot! Looks very professional ;)

Kunne du navigere mellem siderne "Dashboard" og "ECG Live"?

11 svar



Hvordan synes du om brugen af farver på applikationen generelt?

11 svar

et højdepunkt, jeg med det samme lagde mærke til

Fin men det kunne også være nice hvis der var et overordnet farve tema

Farverne er ikke for skrappe og der er samme farvestemning indenfor hver individuel graf (for eks. grøn og mørkegrøn i den første graf)

Farverne er meget professionelt anlagt og de viser en form for identitet. Dog ift. labels, så ville jeg gøre teksten federe (dvs. tykhed), da det vil blive en smule svært at se tydligt med hvid tekst på lyseblå labels.

Super

Tekst farven (Hvid) af titel teksten på jeres plot kan være lidt svær at se. Måske er skriften også lidt lille.

Diagrammerne og menuen til venstre er godt men resten går lidt i et farvemæssigt

Hvordan synes du om brugen af farver på de forskellige grafer?

11 svar

Godt

samme

de spiller godt sammen på de individuelle grafer, men clasher lidt når man sammenligner graferne

Se svar ovenover

Nemt at kende forskel på de forskellige data. Det blænder heller ikke øjnene, dvs. de er satureret rigtigt. Dog ville jeg tjekke om de er venlige for mennesker med farveblindhed.

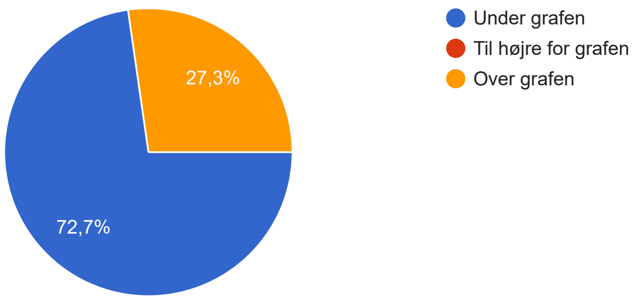
Super

Fint.

Good

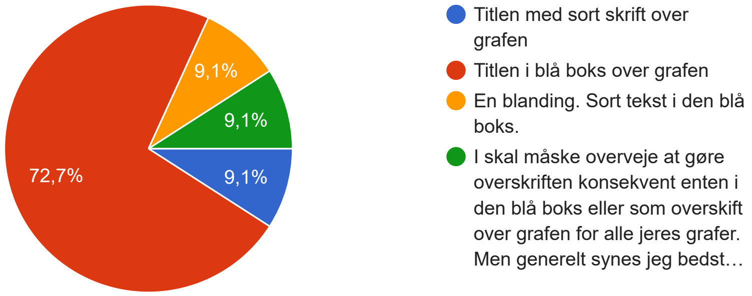
Hvor foretrækker du placering af labels til grafen?

11 svar



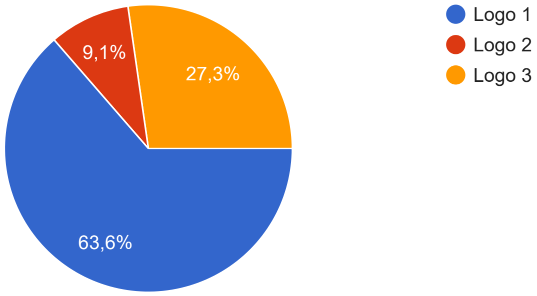
Hvordan foretrækker du grafens titel?

11 svar



Hvilket logo foretrækker du?

11 svar



Har du nogle kommentarer eller forslag til logoet?

6 svar

fattigdom

Man kunne ikke trykke på fanerne "Measurements" og "Notifications", men det var måske heller ikke meningen?

Nej

Nej det er godt

Nope, super cool stuff!

Jeg kan godt lide logoet, og det er klart logo 3 som bedst ligner et stetoskop. Dog kunne I overveje at jeres farve på logoet var sammenhængende med de blå bokse - I kan evt. gøre de blå bokse samme matte lilla farve?
Ellers synes jeg meget godt om overblikket over designet af appen. Men overvej farvevalget (det kan være mere sammenhængende).

A.14.2 Second UX Questionnaire

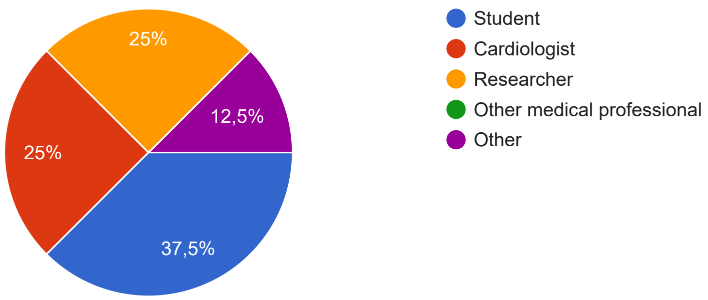
User-friendliness for HeartRater.live

8 svar

Offentliggør analyse

What is your background?

8 svar



Please open the application and play around with it, what's your first impression of the app?

8 svar

Very simple to use and nice design

Very Nice app. Navigation is easy.

Virker simpel

Relatively user friendly. Default Blood pressure and steps should start at 0. There are no denominations on the axes and speed of ECG also missing. Too many numbers on the axes in my opinion

I think it gives a good overview of relevant patient data

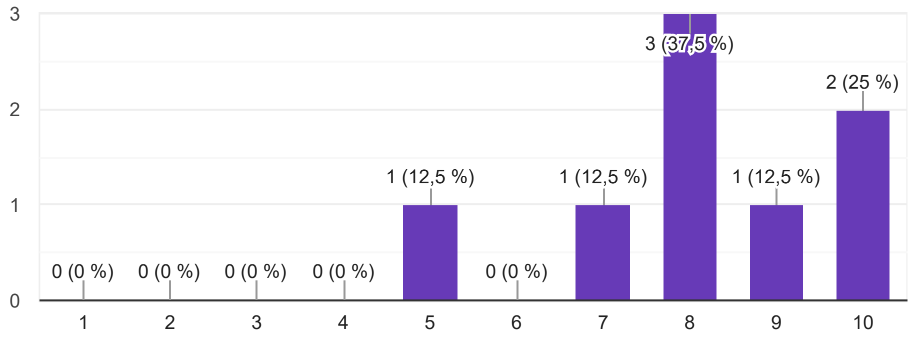
let at forstå

Overskueligt, brugervenligt, nemt at tilgå

It seams nie and easy to get aound

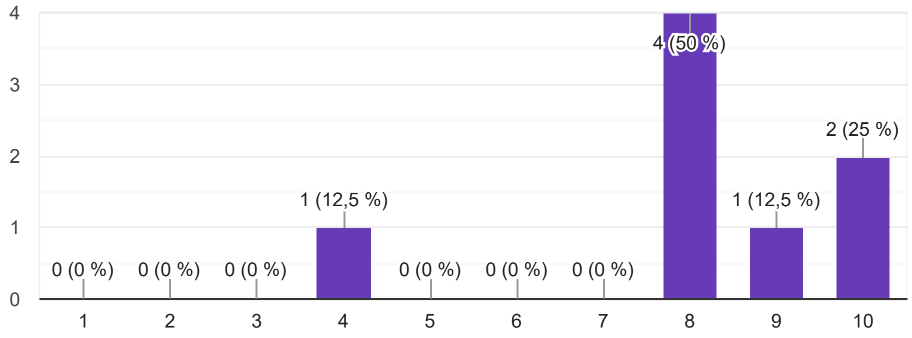
Is the platform intuitive to navigate in?

8 svar



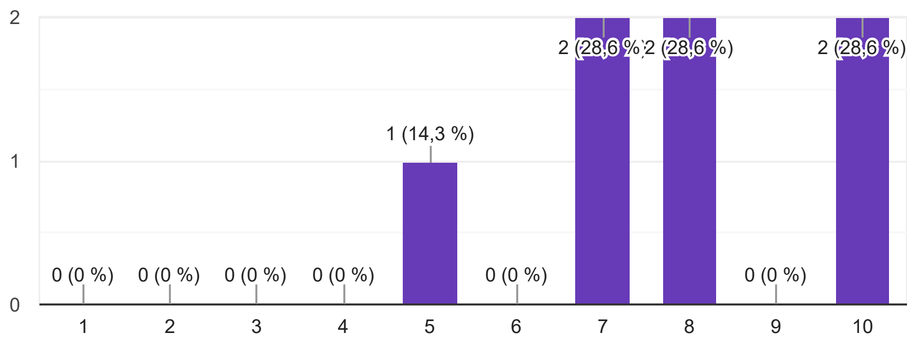
How will you rank the overall design of the platform?

8 svar



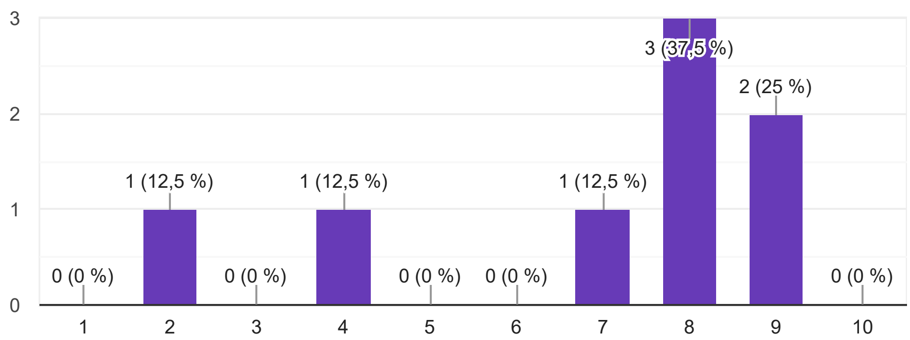
On the Dashboard-page, try to change the dates for the measurements. Now the graphs should change correspondingly if there's data from the specified date. How was your experience doing this?

7 svar



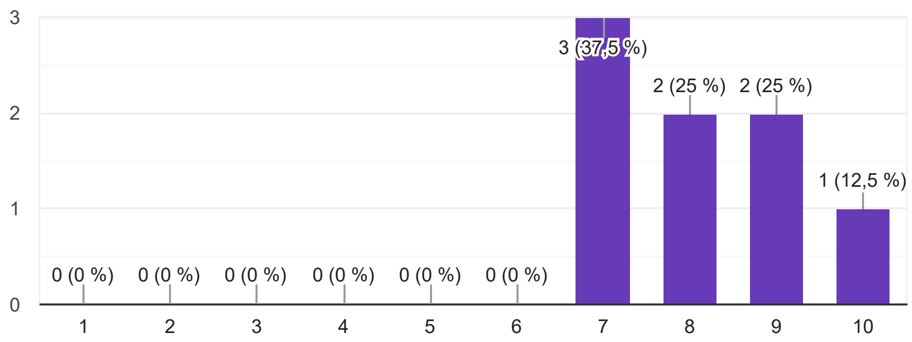
On the Dashboard-page, click the 'Set Thresholds'-button. Now you're able to set limits for every kind of measurement and visulize it on the graph. In the future the platform will be able to send notifications if these thresholds are exceeded. Do you find this feature useful?

8 svar



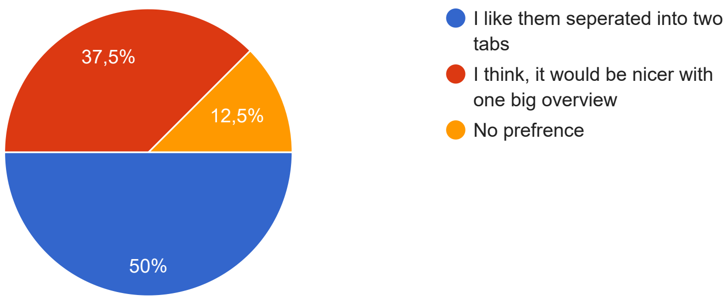
Now go to the Live ECG-tab. You won't see anything on the graph. Now, if you click the 'Get Recorded ECG'-button you'll see some historic ECG-data played in a loop. How was your experience?

8 svar



The Dashboard and the ECG live are on two different tabs, what do you prefer?

8 svar



Do you think you would find this application useful as a cardiologist/medical professional/researcher? Why? Why not?

6 svar

Great

Yes it is useful

Lidt simpel. EKG-data live er meget cool - men et stillestående EKG er der man stiller diagnoser.

Maybe it is not so unique. But if it was part of an equipment we have bought I would be okay with it.

Im a student, so I'm not relly sure here :) But I guess that a medical professional would like an application like this, as it gives them easy access to review the patients data for example before a consultation

I think it could be useful if its posible for a patient to tack it ever single day. So maybe it could be conected to a smartwatch or so.

NOTE: I could not find the Set Thresholds botton on my phone.

A.14.3 Think-Aloud: Validation of eMed app

Evaluation and validation of E-Med

This paper will describe the methods and results found from testing a functional prototype of the mobile application E-Med against Heuristic Evaluation and the usability protocol Think-Aloud.

Heuristic Evaluation

We take our starting point in the 10 Usability Heuristics for User Interface Design developed by Jakob Nielsen[1]

Method

Two UX experts were asked to systematically determine E-Med's usability using the 10 heuristics, furthermore the experts were asked to rate each heuristic from 1 (very poor or lacking completely) to 10 (very good). The experts were handed a mobile phone with the prototype installed.

Results

#1 Visibility of system status

Done well:
- Bottom menu highlights the page currently active (e.g. Home)
- Connection-status to sensor is displayed (online/offline)
- When waiting for the live ECG to display, a loading bar is seen.

Lacking:
- When on the tab 'History' or 'Me' it is unclear if you are connected to an ECG sensor or not.

Rating: 8/10

#2 Match between system and the real world (5)

Done well:
- Icons are intuitive and easy to understand

Lacking:
- Server is internal jargon - we fear that the end-user won't understand it.
- The difference between Sensor and Device is unclear.

Rating: 5/10

#3 User control and freedom

Lacking:

[1] Jakob Nielsen, 1994, Usability Inspection Methods, a summary can be found on the homepage of Nielsen's heuristics: https://www.nngroup.com/articles/ten-usability-heuristics/

- No ‘go-back’, ‘undo’ or ‘cancel’ buttons. E.g. when watching the live ECG, you have to press ‘Home’ again to go back.

Rating: 2/10

#4 Consistency and standards

Done well:

- Color and design is consistent.
- Order of Calendar in steps and sleep are consistent, intuitive order.
- The placement of icons are also intuitive and consistent.

Lacking:

- Device is present under the three dots together with sensor+server, but on home page it is missing.

Rating: 9/10

#5 Error prevention

Lacking:

- In general ‘undo’ is not supported, which means higher chance of error prone systems.
- Error messages are missing if the device is not connected. The user needs some explanation and a suggested action, e.g. a message saying ‘Can’t connect to device, switch on and off your bluetooth and try again’

Rating: 3/10

#6 Recognition rather than recall

Done well:

- Highlighted labels and icons when pushing a button or you are on the current page.

Lacking:

- The offline/online status of the sensor should be visible on all pages.

Rating: 9/10

#7 Flexibility and efficiency of use

Done well:

- Generally done well, especially through the menu bar at the bottom.

Lacking:

- It could be more customizable and thereby flexible by being able to move around the cards that are displayed under the tab ‘History’, such that the user can prioritize which information they want to see first and for which time-period.

Rating: 6/10

#8 Aesthetic and minimalist design

Done well:

- No unnecessary information is displayed and the focus is on the essentials.

Lacking:

- Maybe a little too minimalistic. A lot of empty space on the 'Home' and 'Me' page

Rating: 8/10

#9 Help users recognize, diagnose, and recover from errors

Lacking:

- The user is not guided to help reconnect a device that is not connected.
- Lack of error messages which make it hard to recover from these.

OBS! As the sensor wasn't available when conducting this evaluation, the error-messages and information giving to the user when the sensor is connected weren't evaluated

Rating: (3)/10

#10 Help and documentation

Done well:

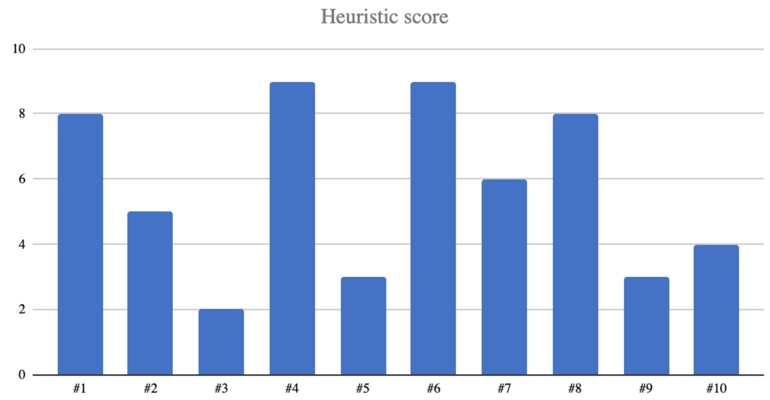
- There is a little help info displayed on the main page under Server and Sensor to explain what they are about.

Lacking:

- There could in general be help icons with tooltip functionality displaying an explanation of various cards.

Rating: 4/10

Heuristic score results:



Conclusion (heuristics)

Overall the prototype is a good basis for a patient application, but it's missing some major parts, especially concerning error prevention, communication of these errors and user freedom. The prototype is doing well with visibility of the system status, consistency, minimizing the user's memory load and minimalistic design.

Think-Aloud

The tests consisted of a combination of exploratory and task instruction.

Method

Three participants were handed the prototype on a mobile phone and asked to imagine that they were now wearing a heart-sensor. The following were asked:

1. Open the application. Explain what you see.
2. Navigate to see your ECG live. What information are you presented with?
3. Go to 'History'. Explain what you see.
4. How many steps did you take on average last year?
5. How many hours of REM, deep and light sleep did you have on the 23rd of June 2021?
6. What do you think about the information presented under the tab 'Me'?
7. What is your general impression of the app?

These were the question guidelines, but sometimes the interviewer would ask additional questions if needed.

Results

In this section we refer to the numbering of the questions above instead of writing all questions again.

Participant 1

1. The screen is pretty empty. I like the information underneath the 'Sensor' and 'Server', but it's still unclear what is meant.
2. It looks fine. Since it's not actually connected I don't really know what to look at.
3. I see steps, sleep, pulse and ECG. The measurements shown are from one day.
4. What does total mean? Like my entire lifetime? Ah total for a year. I think the information about the total steps is irrelevant and confusing. Maybe you could add it somewhere as a sort of fun bonus-info
5. *The phone crashes when wanting to go here*
6. I like the call function, and it even puts the number in your phone. It's also a pretty empty screen. I know this phone is pretty big, but because the information is placed pretty high up on the screen, it's difficult to reach with your thumb.
7. It's maybe a little too minimalistic, it looks like a very standard app, right from the app-design-catalog. I would have liked some of that information about my health displayed somewhere. And why does the patient themselves need to go in and say 'now I want to connect to the sensor', why can't it do it on it's own when I open the app? The app is quite basic, but I could imagine using it when it was further developed. Also I think that the 'Home'-screen should be the 'History' page instead.

Participant 2

1. The design is nice, I like the boxes of information. It's a little unclear what 'server' and 'sensor' is. Underneath the three dots there's a device option as well, I'm not sure what the difference between device and sensor is, or what a device is supposed to mean in the first place.

2. It’s been pending for a while. Looks ok. It’s annoying that there aren’t any ‘go back’ button.
3. The steps and sleep are only showing for the previous day or? I think I would like it better if it showed like a weekly average, that would make better sense, or maybe even a monthly average. Or like, how much better or worse I’m doing compared to last week, that would be kind of cool.
4. That’s a lot of steps. I don’t know why I would need to know how many steps I took in a full year, I mean it’s funny, but not very relevant I guess. Otherwise I think the min, max and average is fine.
5. *The phone crashes when wanting to go here*
6. It looks fine, good information. I like how the boxes can expand and collapse.
7. The app doesn't really explain anything about my health, I would like that, some sort of conclusion. I think the design is okay. Little confused about setting the whole thing up, there could maybe have been some sort of manual.s

Participant 3

1. I see a home screen displaying the option of connecting to Server or Sensor. I am unsure of the difference between the two is, or why I would want to connect to a server. I am expecting to wear a heart sensor and only want to connect to that.
2. I see ECG, pulse and condition. I don’t know what condition means or what I should expect being displayed here. There is a pause feature which was cool, but no return button.
3. I see my steps, sleep, pulse and ecg. I expect the presented data to be from 24 hours, but it is not totally clear.
4. I took 3519. It took me a while to find out that I could click on the steps card to get more info about the steps.
5. I could not go back, when I came from the steps infomation, I had to reclick history. I had to click many-many-many times on the left arrow on the ‘day’ tab to get to the 23rd of june 2021. There was to specification that it was in fact year 2021. But I think about 1.25 h deep, 4.75 h light and 1.5 h rem sleep.
6. Very nice. I have no further comments.
7. It is good. It is missing some design aspects to be completely user friendly, especially back buttons. But generally good and I understood most of it.

Conclusion (Think-Aloud)

From the Think-Aloud test, we got insights about the usability of the App. We can conclude that the app in general is missing some explanatory text or optionally, an onboarding flow for first time users. Also a back-button is needed in all screens to ease the navigability. The developers should be a bit more clear about their end-user and what is relevant for them. A feature like total steps in a year is not useful for the end-user. The Me-page is very well designed, but the other pages could use a bit of redesign to optimize the user experience. But all-in-all it is a useful app that works well for an MVP.

B Selection of antenna for Cellular Internet of Things (C-IoT) tests

The appendix gives an overview of the antenna choices used while performing the Cellular Internet of Things (C-IoT) Key Performance Indicator (KPI) experiments. Choosing the right antenna for the tests was crucial and could significantly affect the performance of the wireless communication technologies. Two main factors were considered when selecting the most suitable antenna for the C-IoT device.

- Antenna matching circuit: This is usually an impedance matching circuit that is designed to ensure the input impedance of the antenna is matched with the output impedance of the Radio Frequency (RF) circuit. A fully matched circuit ensures that the maximum RF power is transferred into the antenna to be radiated.
- Return Loss (s11): It measures the reflected power or the signal strength that bounces back toward the transmission line.

In the case of the off-the-shelve devices, the device manufacturer designed the matching circuit. Therefore, in theory, it was just to find the antenna with matching input impedance to ensure the maximum power is transferred to the antenna. In the case of all the devices used in the C-IoT KPI experiments, were having a 50 Ω. Therefore, all the antennae chosen for the experiment had a match impedance of 50 Ω.

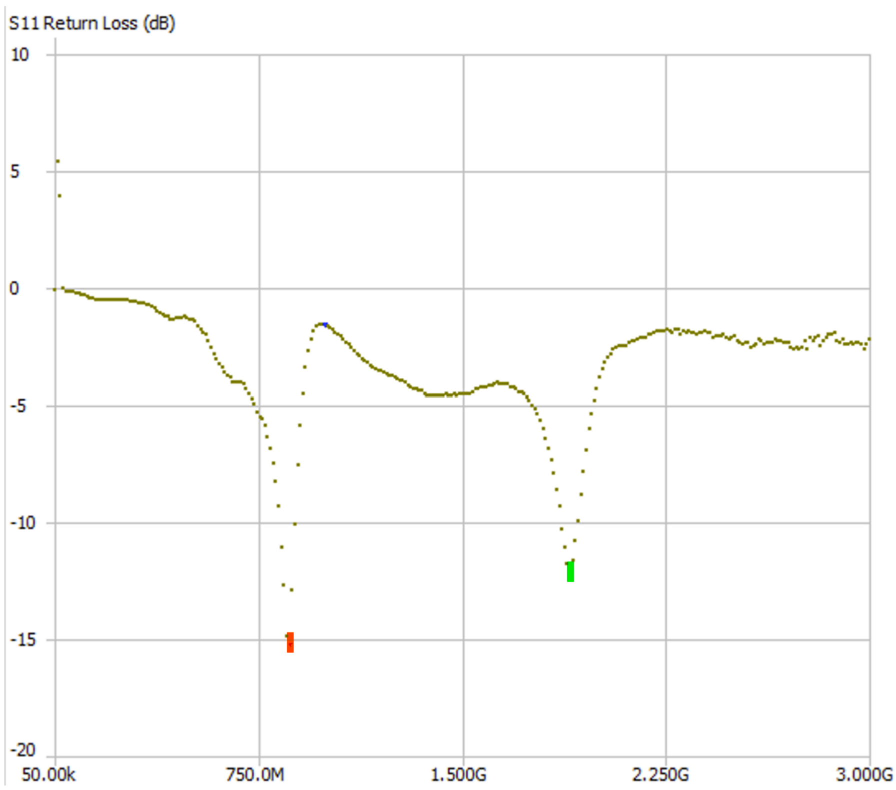


Figure B.1: Example of return loss (s11) graph of one of the chosen antenna [131]

Figure B.1 shows the return loss (s11) graph across different frequencies. The red mark indicated the lowest s11 value of -15dBm at around 868 MHz. The C-IoT network is deployed in LTE band 20 in Denmark, which uses frequencies between 832 MHz – 862 MHz for UPLink (UL) transmission and 791 MHz – 821 MHz DL transmission [132].

In order to further verify the performance of the antenna while connected to Pycom Fipy, Pycom Gpy, and SaRa-R410, multiple such matching antenna (all with 50 Ω impedance) were tested in Line of Sight (LoS) from the Evolved Node B (eNB) on campus and the antenna with the highest Reference Signal Received Power (RSRP) value was chosen for the C-IoT KPI testing. The chosen antenna had the highest RSRP value for both the Mobile Network Operators (MNO).

C Overview of ECG

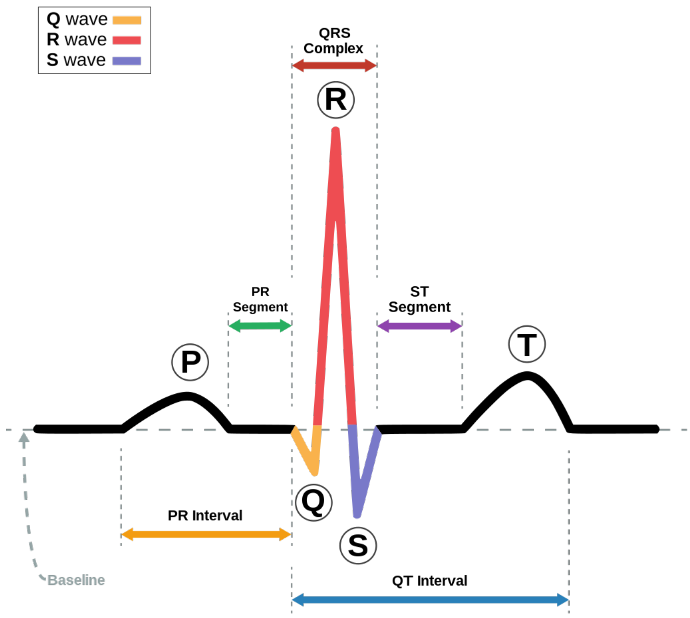


Figure C.1: Waveform annotation in an ECG[133]

Figure C.1 shows the different electrical activity recorded during the single heart beat.

- **P wave:** P wave is caused by the depolarisation of the atria myocardium, which intern triggers the mussel contraction of the heart. This marks as the beginning of the heart beat. Under normal circumstances the duration of the P wave is less than 0.12 seconds and the height is about 0.25 mV.
- **PR interval**: The PR interval is the time taken from the beginning of the P wave till the start of QRS complex. In other words, PR interval shows the time taken by the electrical signal generated by the heart to travel through and finish the mussel contraction, in medical terms it also known as atrial systole. The normal PR interval is between 0.12-0.2 seconds.
- **QRS complex:** The QRS complex represents the depolarisation of the ventricular myocardium and start of the contraction of the large ventricular muscles. In medical terms it is called as ventricular systole. In the case of adults, the duration of the QRS complex is between 0.075 to 0.105 seconds. This is where, the pumping of heartbeat takes place, this is why the electrical signal during this wave is the highest. Heartrate can be calculated using the interval between two R waves.
- **T:** The T wave is when the heart goes from depolarised to polarised stage again, in medical terms where the ventricular repolarisation of the heart takes place. The time between start of Q wave and end of T wave is also called as QT interval. The QT interval, typically shortens as the heartrate increases and the other way around when the heartrate decreases.

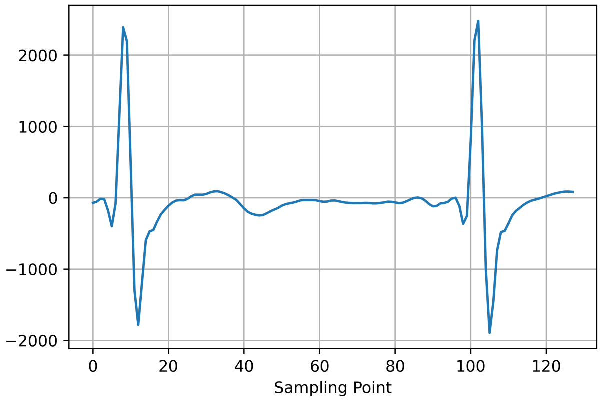


Figure C.2: Regular ECG waveform recorded from the continuous remote monitoring system [78]

Figure C.2 shows the regular ECG data recorded by the continuous remote monitoring system designed during the PhD project.

Figure C.3: Example of an irregular ECG waveform [134]

Figure C.3 shows an example of irregular ECG data.

D Power consumption of the end device

This appendix gives an overview of the power consumption measurement conducted using the nRF PPK2.

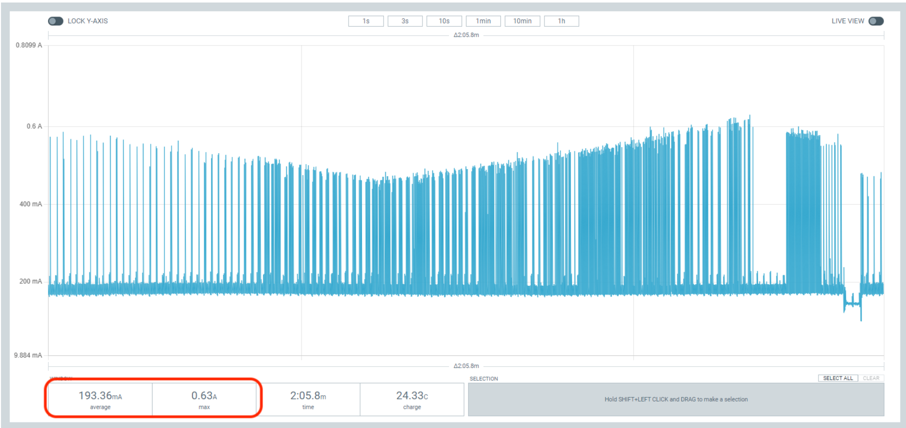


Figure D.1: Power consumption developed Internet of Things (IoT) device

Figure D.1 shows the power consumption of the developed continuous Electrocardiogram (ECG) and Heart Rate (HR) monitoring device when it is in an indoor scenario. The device is monitoring the ECG and HR in a Non Line of Sight (NLoS) location. As can be seen from the figure, the maximum current consumption of the device is 630 mA, whereas the average current consumption of the device is 190 mA over two minutes.

Considering these values in the calculation, the device can be powered continuously for over 25 hours where ECG and HR data is being sent to the remote server. The power calculations consider a 5000 mAh lithium-ion (Li-ion) battery chemistry. This device's power consumption is without any power optimization settings such as Extended Discontinuous Reception (eDRX), Power Saving Mode (PSM), general code optimization, etc., applied to the device.