UNIVERSITY
OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING
DEGREE PROGRAMME IN WIRELESS COMMUNICATIONS ENGINEERING

MASTER’S THESIS

INTEGRATION OF A CELLULAR INTERNET-OF-THINGS TRANSCEIVER INTO 6G TEST NETWORK AND EVALUATION OF ITS PERFORMANCE

Author	Kutay Boz
Supervisor	Dr. Konstantin Mikhaylov
Second Examiner	Dr. Tuomo Hänninen

June 2023

Boz K. (2023) Integration of A Cellular Internet-Of-Things Transceiver Into 6G Test Network and Evaluation of Its Performance. University of Oulu, Faculty of Information Technology and Electrical Engineering, Degree Programme in Wireless Communications Engineering. Master’s Thesis, 34 p.

ABSTRACT

This thesis focuses on the integration and deployment of an aftermarket cellular IoT transceiver on a 6G/5G test network for the purpose of evaluating the feasibility of such device for monitoring the network performance. The cellular technology employed was NB-IoT paired with a Raspberry Pi device as the microprocessor that collects network telemetry and uses MQTT protocol to provide constant data feed.

The system was first tested in a public cellular network through a local service provider and was successfully connected to the network, establishing TCP/IP connections, and allowing internet connectivity.

To monitor network information and gathering basic telemetry data, a network monitoring utility was developed. It collected data such as network identifiers, module registration status, band/channel, signal strength and GPS position. This data was then published to a MQTT broker. The Adafruit IO platform served as the MQTT broker, providing an interface to visualize the collected data.

Furthermore, the system was configured for and deployed on a 6G/5G test network successfully. The device functionality that was developed and tested in the public network remained intact, enabling continuous monitoring and analysis of network data.

Through this study, valuable insights into the integration and deployment of cellular IoT transceivers into cellular networks that employ the latest IoT technology were gained. The findings highlight the feasibility of utilizing such a system for network monitoring and demonstrate the potential for IoT applications in cellular networks.

Keywords: Cellular IoT, Transceiver integration, NB-IoT, 6G, 5G, Raspberry Pi, Network monitoring, cellular network telemetry, MQTT, Network data analysis, Internet of Things applications, IoT, Signal strength telemetry.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

3GPP	3rd Generation Partnership Project
5G	5th Generation
5GTN	5G Test Network
6G	6th Generation
API	Application Programming Interface
C-IoT	Cellular Internet of Things
CoAP	Constrained Application Protocol
E-UTRA	Evolved UMTS Terrestrial Radio Access
EC-GSM-IoT	Extended Coverage GSM IoT
GNSS	Global Navigation Satellite System
GPIO	General Purpose Input/Output
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IoT	Internet of Things
JSON	JavaScript Object Notation
LPWAN	Low-Power Wide-Area Network
LTE	Long-term Evolution
LTE-M1	Long-Term Evolution Machine Type Communication
LTE-NB1	Long-Term Evolution Narrowband
LTE-eMTC	Long-Term Evolution Enhanced Machine-Type Communication
LwM2M	Lightweight Machine to Machine
M2M	Machine to Machine
MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrowband Internet of Things
NR	New Radio
OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
QoS	Quality of Service
SC-FDMA	Single Carrier Frequency-Division Multiple Access
SIM	Subscriber Identification Module
SPI	Serial Peripheral Interface
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRA	UMTS Terrestrial Radio Access

1 INTRODUCTION

The Internet of Things (IoT) has emerged as a transformative technology, enabling a wide range of applications across various industries [1]-[3]. Cellular IoT has gained significant attention due to its ability to provide reliable and ubiquitous connectivity for IoT devices [4], [5]. In this thesis, we focus on the integration and deployment of an aftermarket cellular IoT transceiver on a 6th/5th Generation (6G/5G) test network [6] and evaluating its viability as a platform for network telemetry provider.

Cellular networks offer several advantages for IoT deployments, including wide coverage, high reliability, and security features. The integration of cellular IoT capabilities with existing devices opens up new possibilities for connected applications [4], [5]. Our study explores the potential of integrating an aftermarket cellular IoT transceiver that employs Narrowband Internet of Things (NB-IoT) technology with a commonly used microprocessor-based system. This has been achieved by implementing the Quectel BG96 LTE Cat M1/NB1 (Long-Term Evolution Machine Type Communication / Long-Term Evolution Narrowband) module [7] with a Raspberry Pi device, which is a small single-board computer, that is able to leverage the capabilities of the module.

The primary goal of this thesis is to develop a robust system that seamlessly integrates the aftermarket transceiver with the microprocessor into a fully functional IoT device, serving as a network logger/monitor for new generation cellular networks using the NB-IoT technology. To achieve this, testing and experimentation are conducted in both a public cellular network and the designated 6G/5G test network in the University of Oulu premises [6]. The system aims to collect network information, including multiple different network identifiers, band and channel info, Global Positioning System (GPS) location, and signal strength measurements, to enable real-time monitoring and analysis of the cellular network environment.

In addition, the system incorporates Message Queuing Telemetry Transport (MQTT) functionality to publish the collected network telemetry to a designated MQTT broker. The Adafruit IO [8] platform serves as the chosen broker as on top of its basic functionality as a MQTT broker, it allows for the visualization and monitoring of the data through its user interface.

The objectives of this thesis can be summarized as follows:

1. Integrate an aftermarket cellular IoT transceiver into a working microprocessor system.
2. Establish reliable connectivity to the designated 6G/5G test network.
3. Develop a network monitoring utility that employs MQTT to publish the collected data.
4. Visualize the collected telemetry data.

The structure of this thesis is as follows:

- In Chapter 2, the technical background for the thesis work is compiled.
- Chapter 3 presents the related work in the field of cellular IoT integration.
- Chapter 4 focuses on the design and implementation details of the system.
- Chapter 5 presents the discussion of the end result of the implementation and its implications.
- Finally, Chapter 6 will provide a comprehensive summary and discuss the implications of the research findings.

2 TECHNICAL BACKGROUND

2.1 Internet of Things

The Internet of Things is a worldwide framework that facilitates advanced services by connecting physical and virtual objects using current and developing interoperable information and communication technologies. It integrates various technologies such as machine-to-machine communication, autonomic networking, data mining, decision-making, security, privacy protection, and cloud computing.

In the context of the IoT, “things” refer to objects in the physical or virtual world that can be identified and integrated into communication networks. Physical things can be sensed, actuated, and connected, including examples like the environment, industrial robots, and electrical equipment. Virtual things exist in the information world and can be stored, processed, and accessed, such as multimedia content and application software. Figure 1 represents an example visualization of an Internet of Things environment.

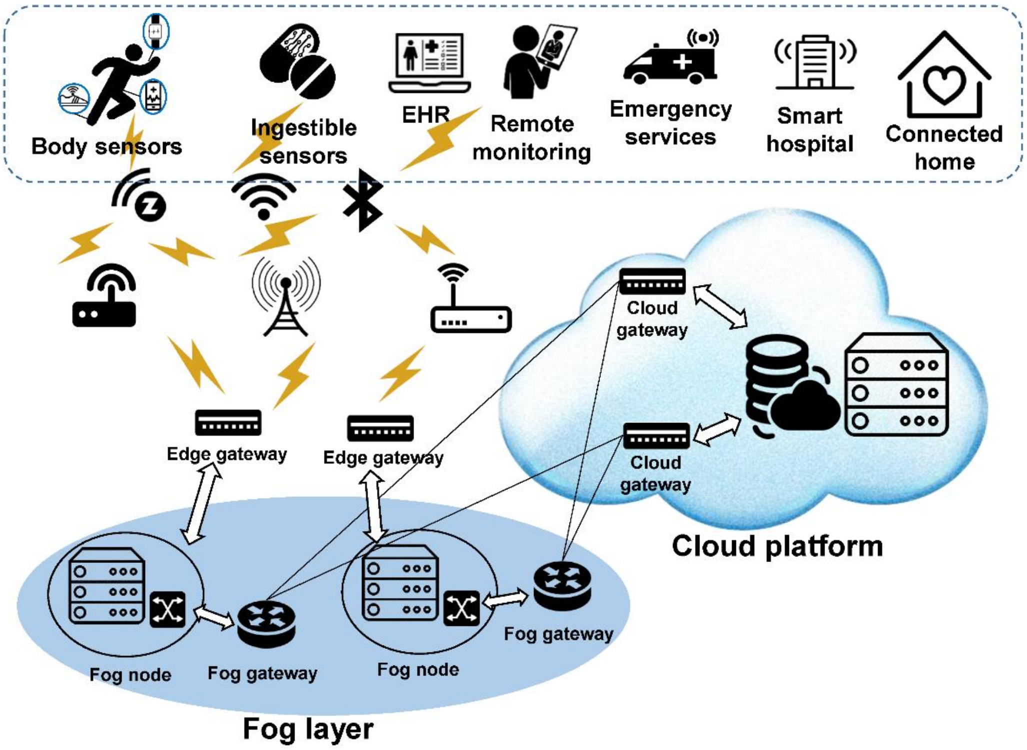


Figure 1. An example overview of an IoT system for healthcare application [9].

Devices in the IoT have communication capabilities and may also have sensing, actuation, data capture, storage, and processing capabilities. They collect information and communicate with other devices through communication networks, either via a gateway, directly, or through a combination of both. The IoT applications cover various areas such as intelligent transportation systems, smart grids, e-health, and smart homes. Communication networks transfer data between devices and applications, providing reliable and efficient data transfer. The IoT network infrastructure can utilize existing networks like Transmission Control Protocol/Internet Protocol (TCP/IP)-based networks or evolving networks like next-generation networks. Devices in the IoT are categorized as data-carrying devices, data-capturing devices, sensing and actuating devices, and general devices, each serving different purposes in connecting physical things to communication networks and facilitating interactions [10].

2.2 Cellular IoT and NB-IoT

Machine-to-machine (M2M) communication was seen as an opportunity for ecological growth by 3rd Generation Partnership Project (3GPP). Low-Power Wide-Area Network (LPWAN) technology, known for its low cost, low power consumption, wide coverage, and low data rates, played a significant role in the era of the Internet of Everything. 3GPP conducted research on cellular networks since 2005, focusing on M2M services and their integration into 5G networks. Over the years, 3GPP addressed issues such as overload, congestion, and resource shortage. In Release 13, 3GPP introduced narrow-band air interfaces, including NB-IoT, which drew industry attention due to its software upgrading and core network reusing capabilities. LTE-M and NB-IoT are optimized for lower complexity, power consumption, deep coverage, and higher device density, expanding the LTE technology portfolio to support the 5G New Radio (NR). LTE-M is suitable for medium and low-rate scenarios, while NB-IoT is a next step towards an optimized network, specifically designed for low-data-rate applications. As can be seen in Figure 2, both technologies are widely adopted globally. Also, both of them offer massive connectivity, low power consumption, and wide area coverage. NB-IoT in particular has promising potential and is supported by a robust cellular communication network [4].

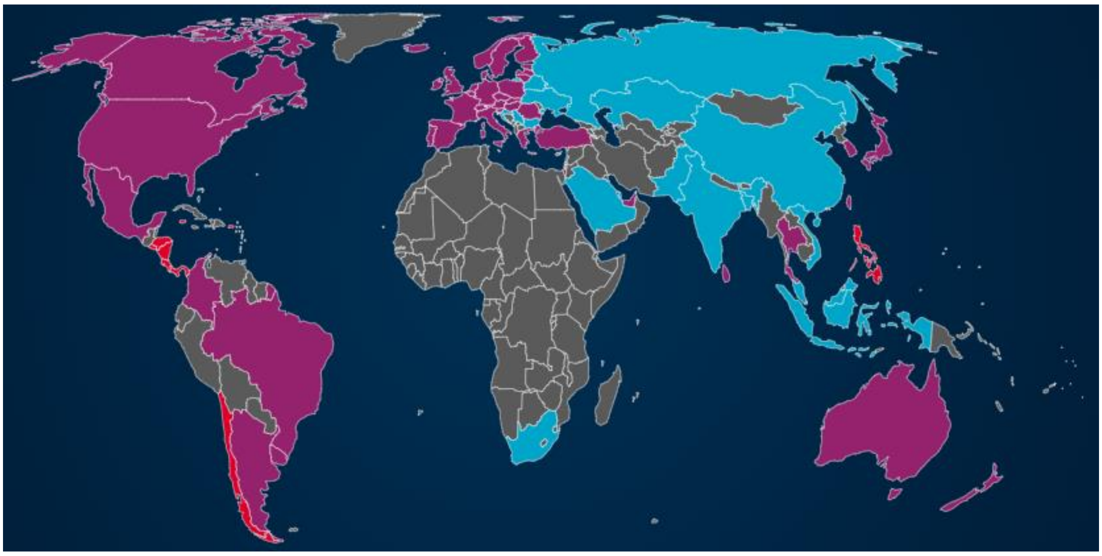


Figure 2. Cellular IoT deployment map. Blue represents NB-IoT only while red is LTE-M only. Purple colour represents the deployment of both technologies [11].

As the Internet of Things (IoT) is continually expanding, consisting of networks with different design objectives, including local area and wide-area coverage. The Third Generation Partnership Project (3GPP) has introduced key features for IoT in Release 13, including Extended Coverage GSM IoT (EC-GSM-IoT) and Long-Term Evolution Enhanced Machine-Type Communication (LTE-eMTC), which enhance existing Global System for Mobile Communications (GSM) and LTE networks for IoT applications. Narrowband IoT (NB-IoT) is another 3GPP technology designed for ultra-low-end IoT applications. NB-IoT is not fully backward compatible but coexists well with GSM and LTE. It can be deployed in various ways, such as replacing a GSM carrier or utilizing a portion of LTE resources with either in-band or guard band configuration as in Figure 3. NB-IoT extensively reuses LTE design, reducing development time and facilitating the adoption of NB-IoT products.

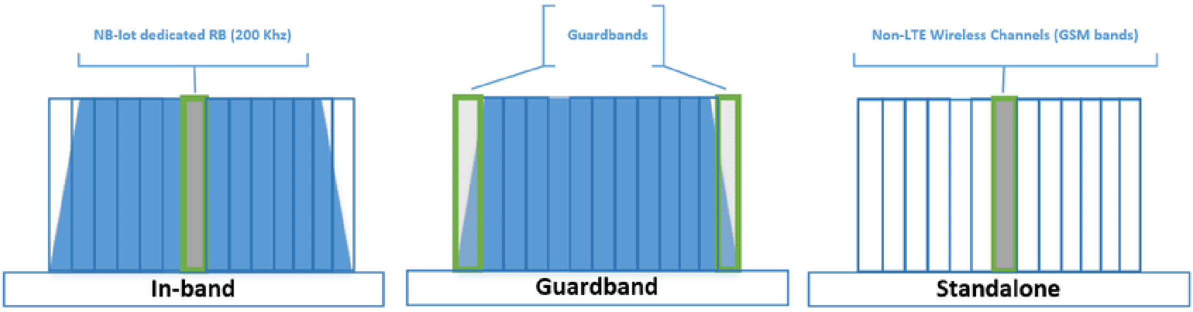


Figure 3. NB-IoT band operation modes [12].

NB-IoT is designed to achieve coverage extension, low device complexity, long battery life, and support for a massive number of IoT devices. Ongoing enhancements in 3GPP aim to introduce features like low-complexity multicast functionality and improved positioning accuracy. NB-IoT serves as a step toward the development of 5G radio access technology for machine-type communications, and it is crucial to ensure its coexistence with future 5G networks. By designing NB-IoT to meet 5G requirements, the need for a separate 5G IoT technology can be minimized, avoiding market fragmentation and leveraging economies of scale. NB-IoT enables ultra-low-cost devices and has the capacity to support a large number of devices in a cell, contributing to the emergence of the fog network where these devices collectively offer significant sensing, intelligence, storage, and computing capabilities [13].

2.3 Signal Level and Quality Measures

Signal strength indicators and measurements provide valuable insights into the quality and strength of the cellular signal received by a device. They help assess the network conditions and the device's ability to establish and maintain a reliable connection. There are several indicators commonly used in cellular networks. Measurements that are related to our study are:

- **Received Signal Strength Indicator (RSSI):** It is calculated as the linear average of the total received power within the configured measurement bandwidth and specific Orthogonal Frequency-Division Multiplexing (OFDM) symbol. This power includes contributions from various sources, such as co-channel serving and non-serving cells, adjacent channel interference, thermal noise, and other factors. The RSSI reflects the overall received power observed by the User Equipment (UE) from all these sources [14].
- **Reference Signal Received Power (RSRP):** It is the linear average of the power contributions (measured in Watts) from the resource elements that carry cell-specific reference signals within the specified measurement frequency bandwidth. When determining RSRP, it is required to utilize the cell-specific reference signals R0 as specified in TS 36.211 [14], [15].
- **Signal-to-Interference-plus-Noise Ratio (SINR):** Calculated as the linear average of the power contributions (measured in Watts) from the resource elements carrying cell-specific reference signals, divided by the linear average of the noise and interference power

contributions (again measured in Watts) from the resource elements carrying the same cell-specific reference signals within the designated frequency bandwidth. When determining the SINR, it is required to utilize the cell-specific reference signals R0 as specified in TS 36.211 [14], [15].

- **Reference Signal Received Quality (RSRQ):** It is defined as:

$$RSRQ = \frac{N*\text{RSRP}}{\text{E-UTRA Carrier RSSI}} \tag{1}$$

where N represents the number of resource blocks in the measurement bandwidth of the Evolved UMTS Terrestrial Radio Access (E-UTRA) carrier RSSI. Both the numerator and denominator measurements should be conducted over the same set of resource blocks.

The E-UTRA Carrier Received Signal Strength Indicator (RSSI) is calculated as the linear average of the total received power (measured in Watts) observed specifically in certain OFDM symbols of measurement subframes, within the measurement bandwidth, over N resource blocks by the UE. The power contributions include signals from various sources such as co-channel serving and non-serving cells, adjacent channel interference, thermal noise, and others [14].

- **Bit Error Rate (BER):** BER is a metric that quantifies the accuracy of data transmission by measuring the number of bits received incorrectly compared to the total number of transmitted bits. It reflects the level of signal distortion or interference affecting the data transmission. BER is typically expressed as a ratio or percentage.

Accurate assessment of signal strength indicators is vital for optimizing network performance, ensuring reliable connectivity, and troubleshooting issues related to signal quality and coverage.

2.4 MQTT

Message Queuing Telemetry Transport (MQTT) is a publish/subscribe messaging protocol designed for lightweight Machine-to-Machine (M2M) communication in constrained networks. It was introduced in 1999 by IBM and Arcom Control Systems Ltd.

MQTT allows clients to publish messages to a broker and subscribe to topics to receive those messages. It is a binary protocol with a fixed 2-byte header and supports small message payloads up to 256 MB. Transmission Control Protocol (TCP) is used as the transport protocol, and Transport Layer Security / Secure Sockets Layer (TLS/SSL) can be employed for security. MQTT offers three levels of Quality of Service (QoS) for reliable message delivery. It is particularly suitable for large networks of small devices that need to be monitored or controlled from a central server on the Internet. However, it is not designed for device-to-device transfer or multicast data dissemination, and it provides limited control options compared to other messaging protocols [16].

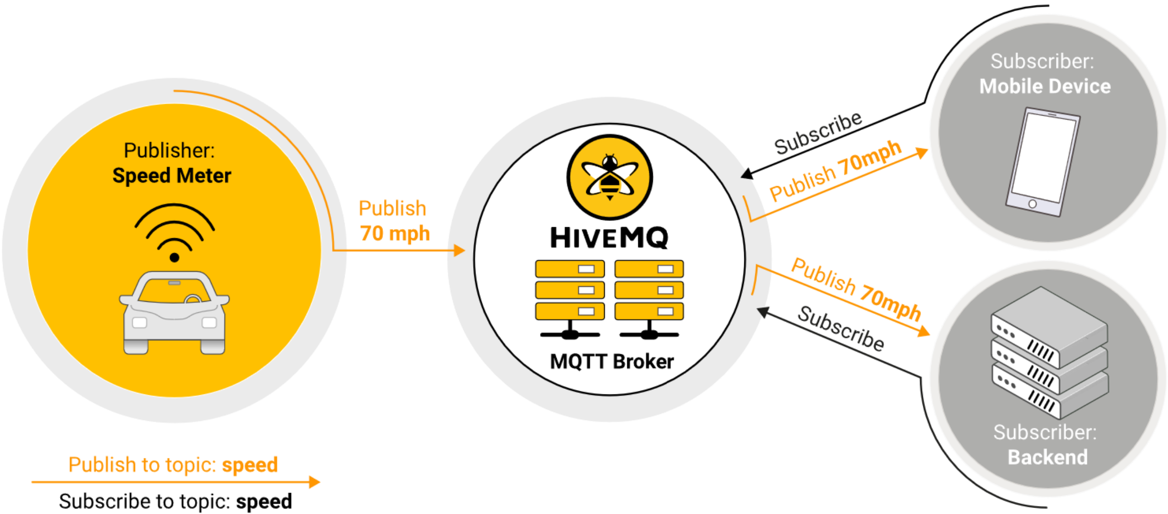


Figure 4. A simple example to a MQTT architecture [17].

MQTT utilizes a messaging model based on topics and subscriptions. Topics are strings to which messages are published and subscribed. They are hierarchical, allowing multiple levels separated by slashes. Clients can subscribe to topics of interest, indicating their desire to receive messages published to those topics. The broker maintains the subscriptions and forwards relevant messages to the subscribed clients. In Figure 4, an example MQTT architecture for vehicle telemetry system can be seen with the mentioned elements. MQTT also supports wildcards, including the single-level wildcard (+) and the multi-level wildcard (#), enabling subscriptions to multiple matching topics. This model offers flexibility and scalability in message handling [17].

2.5 University of Oulu 5G Test Network

The 5G Test Network (5GTN) facilities in Oulu, Finland provide exceptional opportunities for real-time testing of 5G technology, components, and new services. Located at the University of Oulu and VTT Technical Research Centre of Finland, these closely situated premises cater to the rigorous requirements of 5G trials. Whether you are testing your 5G technology, components, or exploring innovative services, the 5GTN in Oulu offers a unique testing environment.

The 5G Test Network (5GTN) is a globally expansive and open-access network for testing 5G technologies. It offers a comprehensive testing system, covering infrastructure, applications, and services, enabling a controlled environment for testing everything from prototype devices to complete solutions. The network spans multiple sites and is well-connected to national and global test environments. With support for various frequency bands, orchestration functionalities, and system testing tools, 5GTN provides a wide range of testing capabilities. It invites users to leverage cutting-edge equipment and data for their innovations, conduct application and technology testing over the live network, and bring their innovative ideas to the open development platform. The network also facilitates research projects aimed at developing specific 5G technology components and services. It also offers advanced analytics for comprehensive trial experiences and outcomes [6].

2.6 GNSS

Satellite positioning, also known as Global Navigation Satellite System (GNSS), is a crucial technology used for precise positioning and navigation. It relies on signals transmitted by satellites to determine a receiver's location anywhere on Earth with an accuracy of a few meters and provides time information with high precision. The primary satellite positioning systems are GPS (US), GLONASS (Russia), Galileo (Europe), and BeiDou (China). These systems consist of satellites equipped with atomic clocks that transmit signals containing time information and auxiliary data.

To calculate the receiver's position, the difference between the transmission time information from the satellites and the reception time from the receiver's internal clock is measured. By multiplying this time difference by the speed of light, the distance between the receiver and each satellite can be determined. The receiver's clock deviation relative to the satellites' clocks must also be considered, and at least four satellites are required to solve for the receiver's position accurately.

Satellite positioning has various applications beyond navigation. It is essential for precise timekeeping, synchronizing devices, and maintaining the timing of critical systems like power plants and mobile phone networks. However, GNSS receivers do not automatically reveal their location to system administrators or others, ensuring privacy and security.

The accuracy of satellite positioning is influenced by multiple factors. Inaccuracies can arise from imperfect satellite orbit and clock predictions, signal distortion caused by the ionosphere and troposphere in the atmosphere, and signal attenuation and reflections due to obstacles like buildings and trees. These factors, along with the weak strength of satellite signals, contribute to positioning inaccuracies. While the typical accuracy for consumer-grade GNSS receivers is around five meters, it can be lower under certain conditions and significantly affected by satellite failures or environmental distortions.

Stricter accuracy requirements are met through assistance services provided by base stations with precisely known locations. These services generate real-time assistance information that improves positioning accuracy, sometimes achieving centimeter-level accuracy. Satellite-Based Augmentation System (SBAS) is another type of assistance service that operates through satellites in geostationary orbits, ensuring reliability for critical applications such as air traffic.

The maintenance and development of satellite positioning systems involve various entities. GPS, initially developed for military use, is maintained by the US Department of Defense. GLONASS was created as a Soviet counterpart to GPS but faced operational challenges before being restored. Galileo is a civilian-operated GNSS system designed to be compatible with GPS, and BeiDou serves the Chinese region. In addition to these global systems, there are also local systems like QZSS (Japan) and Navic (India).

Overall, satellite positioning plays a vital role in modern positioning and navigation, providing accurate location information worldwide. Assistance services, advancements in receiver technology, and the compatibility between different satellite systems contribute to improving accuracy and reliability, benefiting various applications in sectors such as transportation, surveying, and telecommunications [18].

2.7 JSON

JavaScript Object Notation (JSON) is a lightweight, language-independent data format that is easy to read and write for humans and easy to parse and generate for machines. It is based on a subset of the JavaScript programming language and is widely supported by various programming languages. JSON is composed of two structures: a collection of name/value pairs and an ordered list of values. Objects are unordered sets of name/value pairs, while arrays are ordered collections of values. Values can be strings, numbers, true/false, null, objects, or arrays, and these structures can be nested. JSON allows whitespace between tokens and follows specific encoding rules [19].

2.8 QoS

Quality of Service (QoS) in networking refers to the use of mechanisms or technologies to control network traffic and ensure the performance of critical applications with limited network capacity. It allows organizations to prioritize high-performance applications and optimize overall network traffic. QoS is commonly used for resource-intensive services like IPTV, online gaming, video streaming, videoconferencing, and VoIP. By implementing QoS, organizations can manage and engineer their network traffic, ensuring expected service quality and user experiences.

QoS works by marking packets to identify service types and creating separate virtual queues for each application based on priority. It helps in managing bandwidth, reducing delay, preventing loss and jitter, and improving overall network performance. QoS is important for guaranteeing the high performance of critical applications, especially those sensitive to latency and packet loss. It becomes increasingly significant as network requirements evolve due to the growing number of users and the emergence of IoT. Techniques and best practices involved in QoS include prioritization, resource reservation, queuing, and traffic marking. The advantages of deploying QoS include application prioritization, better resource management, enhanced user experience, point-to-point traffic management, packet loss prevention, and latency reduction. Fortinet has been recognized as a leader in SD-WAN, a technology that enables QoS implementation and extends high-performance networks across branch offices [20].

2.9 UART

Universal Asynchronous Receiver/Transmitter (UART) is a widely used device-to-device hardware communication protocol. UART allows for the transmission and reception of serial data using only two wires. UART operates by transmitting data in the form of packets, which consist of a start bit, data frame, parity bit, and stop bits. It is important to understand and properly implement the UART for developing robust and high-quality products. The use cases of UART includes debugging, manufacturing function-level tracing, customer/client updates, and testing/verification [21].

UART is a simple protocol used for exchanging serial data between devices. It operates with only two wires and can transmit and receive data in both directions. UART can be simplex, half-duplex, or full duplex. It is commonly used in older devices with RS-232 interfaces but has been replaced by protocols like SPI and I2C in many modern applications. UART requires timing synchronization, and the baud rate, frame structure, and parameters must match on both ends of the connection.

A UART frame consists of start and stop bits to indicate the beginning and end of user data. Data bits carry the actual information, and there can be 5 to 9 bits in a frame. Parity bit, an optional element, helps detect single bit errors. UART is still widely used for low-speed and low-throughput applications due to its simplicity and cost-effectiveness [22].

3 RELATED WORKS

In this section, we discuss previous works that have addressed similar problems and provide insights into the solutions for NB-IoT based monitoring and telemetry systems. Understanding the existing body of knowledge will help us identify gaps and opportunities for our research.

The study in [23] proposes a methodology for sustainable farming irrigation using a combination of Wireless Sensor Networks (WSN), Narrowband Internet of Things (NB-IoT), and machine learning. The goal is to monitor and control farming irrigation to avoid overirrigation and alert to risk situations. The system consists of a WSN installed on the ground, a main server running machine learning algorithms, and an iOS application for remote monitoring and control. Communication is established using NB-IoT and LoRa, and data processing is done using machine learning algorithms. The system aims to improve water management in agriculture and promote sustainability. The paper also discusses related work, the system architecture, hardware and software components, and implementation scenarios.

There is also another thesis work from Aalto University [24] which is a study that aligns quite well with our aim. It focuses on the development of an embedded system utilizing Narrowband Internet of Things (NB-IoT) as the radio interface for low-cost and low-power data transmission in IoT applications. The Lightweight Machine to Machine (LwM2M) protocol is chosen as the communication protocol. The system utilizes the STM32L476 microcontroller. The study is able to provide valuable information about the feasibility of a similar system that we intend to build employing NB-IoT technology for end-users or customers.

The paper [25] discusses the communication protocols suitable for low-power massive IoT devices. The authors analyse and compare protocols such as TCP, User Datagram Protocol (UDP), Constrained Application Protocol (CoAP), and Message Queuing Telemetry Transport (MQTT) in the context of data transmission in massive IoT scenarios. The focus is on the implementation issues and efficient data transmissions with respect to Quality of Service (QoS) requirements. The practical implementation is done using the NB-IoT standard. The paper provides a side-by-side comparison of the protocols' overheads and their impact on data usage. The key contributions of the paper include an analysis of available IoT communication protocols, evaluation of NB-IoT technology, calculation of expected data consumption, and lessons learned from the assessment of IoT protocols for NB-IoT. The paper provides insights into the protocols and their suitability for low-power massive IoT applications.

In [26], the paper describes the design of an air quality monitoring system based on NB-IoT technology. The system utilizes an STM32F103RCT6 microprocessor unit as the core processor and collects data from gas pollutants using an electrochemical gas sensor. The system is capable of wireless network communication through NB-IoT or General Packet Radio Service (GPRS) modules, allowing for stable data transmission. In case of network abnormalities, the system can automatically switch to the alternative network or store the data in Serial Peripheral Interface (SPI) flash memory for later retrieval. The overall design of the system includes the perception layer (sensors), transmission layer (NB-IoT and GPRS modules), and control layer (STM32F103RCT6). The paper also discusses the hardware design, including the master chip module, power module, information acquisition module, and wireless communication module. The software design involves data acquisition, transmission, time calibration, and uploading data to a cloud platform. The system enables monitoring of air quality in special field environments and allows users to retrieve air quality data from the cloud platform.

The paper in [27] investigates the impact of different messaging protocols on the performance of NB-IoT smart meters. The study focuses on two protocols, MQTT and CoAP, and evaluates their effect on signal quality, network registration, and packet losses of SIM7020E modem used in NB-IoT smart meters. The experiments were conducted in various environments, including indoor, outdoor, and inside a polycarbonate box in a building basement. The research aims to provide insights for developers in selecting the optimal transport protocol for NB-IoT smart meters based on factors such as data losses, signal strength, power consumption, and latency. The results of the experiments are discussed, including packet loss rates, Received Signal Strength Indication (RSSI), and the suitability of MQTT and CoAP protocols for NB-IoT smart metering applications.

[28] presents a dairy cow health monitoring system based on NB-IoT communication. The proposed system aims to detect unusual changes in cows' physical status, such as body temperature and exercise capacity and notify the administrator in real-time. The system utilizes a monitor worn on the cow's neck to record physical parameters, which are then transmitted to a server via NB-IoT communication modules. An Android application is used to analyze and process the data. The hardware design includes components such as the STM32 microprocessor for low-power operation, BC-95 communication module for NB-IoT communication, ADXL345 accelerometer for collecting movement data, and DS18B20 temperature sensor for measuring body temperature. The paper also discusses the use of Constrained Application Protocol (CoAP) for IoT device communication and the configuration of the OceanConnect platform for IoT ecosystem management. Overall, the proposed cow health monitoring system based on NB-IoT communication offers advantages such as wider coverage, reduced power consumption, and lower cost compared to similar products in the market.

The paper [29] discusses the design of a monitoring system for industrial and environmental applications using Internet of Things (IoT) technology. The paper focuses on the connectivity aspect of IoT and evaluates the performance of two protocols, CoAP and MQTT, over Narrowband-IoT (NB-IoT) networks. It also proposes a new algorithm for data clustering to improve data processing and visualization for monitoring solutions in industrial and environmental settings. The paper suggests that IoT can provide a flexible and scalable solution for monitoring environmental hazards by connecting various devices to the internet and sharing information among them. The paper discusses the NB-IoT communication standard and its suitability for IoT applications. It explains the air interface and MAC design of NB-IoT, including the use of Orthogonal Frequency-Division Multiple Access (OFDMA) and Single Carrier Frequency-Division Multiple Access (SC-FDMA) for downlink and uplink transmission. The calculation of data rates in the IoT system is also presented. Furthermore, the paper compares the performance of CoAP and MQTT protocols in terms of the number of transport blocks and achievable throughput. The calculations provide insight into the data transmission efficiency of both protocols. In conclusion, the paper emphasizes the importance of IoT in designing monitoring systems for industrial and environmental applications. It highlights the benefits of NB-IoT, CoAP, and MQTT in enabling connectivity and efficient data transmission. Overall, the paper provides insights into designing an IoT-based monitoring system, evaluating protocols, and proposing a data clustering algorithm for industrial and environmental applications.

The paper in [30] discusses the implementation and architecture of a smart street lighting system using Narrowband Internet of Things (NB-IoT) technology. The paper presents the system's architecture, which consists of four layers: perception and control layer, transport layer, platform layer, and application layer. The perception and control layer includes sensors and controllers installed on individual streetlights, which collect data and communicate through the NB-IoT network. The transport layer utilizes the NB-IoT communication network provided by China Telecom for data transmission. The platform layer incorporates the IoT management platform Huawei OceanConnect for connection management, device management, data calculation, and remote-control functionalities. Finally, the application layer encompasses various services and applications enabled by the smart street lighting system. The authors emphasize the advantages of the smart street lighting system based on NB-IoT, such as advanced communication technology, flexible control strategies, real-time monitoring, and efficient management. They explain that the system enables independent control of each streetlight, adjusts brightness based on illumination and traffic conditions, and integrates additional urban management services like wireless communication, video surveillance, and environmental monitoring. The paper also discusses the implementation of an actual smart street lighting system using STM32 for perception and control, China Telecom's NB-IoT network for data transmission, and Huawei OceanConnect for platform functionalities. The system's feasibility is verified, demonstrating its potential to provide intelligent, efficient, and cost-effective services for smart cities. The paper highlights the benefits of implementing a smart street lighting system based on NB-IoT technology and provides insights into its architecture and practical implementation.

Authors in [31] discusses the application of Narrowband Internet of Things (NB-IoT) in intelligent fire protection systems. It highlights the advantages of NB-IoT technology in enhancing the combat capability of fire-fighting forces, preventing fires, and reducing loss of life and property. The paper focuses on the use of NB-IoT technology in smoke-fire detection and alarm devices. It identifies the shortcomings of independent smoke products and proposes solutions to address them. The paper also describes the characteristics and solutions of NB-IoT in smoke sensing, including the terminal layer (Internet of Things sensor fusion), network layer (NB-IoT deployment with wide coverage), platform layer (unified platform multi-service convergence management), operating layer (enriching smoke sensors), and service layer. It highlights the advantages of NB-IoT technology, such as low power consumption, real-time monitoring, automatic checking, and multi-form push of real-time information. The paper concludes by discussing the development and prospects of NB-IoT technology in intelligent solutions, emphasizing its value in improving urban operations, production efficiency, and consumer quality of life.

The paper [32] presents the design and implementation of an intelligent manhole cover monitoring system based on Narrowband Internet of Things (NB-IoT) technology. The proposed system consists of a perception layer, network layer, and application layer. The sensing layer utilizes an embedded ARM microprocessor and sensor technology to collect data from manhole covers. The network layer employs NB-IoT technology for communication between the access device and the background server. The application layer handles data storage, display, and alarm functions. The system aims to enable intelligent management of manhole covers in smart cities, contributing to a safer and more harmonious urban environment. The paper discusses the system structure, hardware design, and software design. The hardware design includes the selection of the main controller (STM32F103C8T6), sensor modules, and NB-IoT communication modules. The software design involves programming the main controller, serial port communication, and the management platform software developed using Visual Studio and Java. The mobile terminal, designed for inspectors, operates on the Android system and facilitates data reception, positioning, navigation, and data interaction with the server. The paper also highlights the advantages of NB-IoT technology, such as wide coverage, low power consumption, and high security. The system has been field-tested and proven to be stable and reliable. In conclusion, the proposed intelligent manhole cover monitoring system based on NB-IoT offers a solution to the challenges faced in managing urban manhole covers.

It enhances safety, enables intelligent management, contributes to smart city development, and establishes a foundation for big data analysis and intelligent decision-making systems.

The paper in [33] presents a slope disaster monitoring and early warning system based on 3D- Micro-Electromechanical System (MEMS) and Narrowband Internet of Things technologies. The conventional methods of slope disaster monitoring have limitations in terms of power consumption and accuracy of warning indexes. To address these issues, the proposed system incorporates 3D-MEMS acceleration sensing for data acquisition and NB-IoT transmission technology for data transmission. The system consists of four subsystems: intelligent sensor acquisition, data transmission, photovoltaic self-powered, and data reception and analysis. The system has successfully demonstrated its ability to collect and transmit multi-source monitoring data accurately. The system provides a new data transmission scheme and multi-source early warning indexes acquisition method for slope disasters. The data transmission subsystem utilizes NB-IoT technology for low-power and cost-effective wireless communication. The data analysis subsystem manages and analyses the collected data, allowing real-time monitoring and timely warning of slope disasters.

The paper [34] proposes a design for a smart car parking system based on Narrowband Internet of Things technology. The system is commanded by a software application that automates the parking and unparking of cars by instructing the number of cars to be parked in assigned parking lots. NB-IoT is chosen as the communication technology for the proposed system due to its low power consumption, wide area coverage, and high connectivity. It provides reliable network construction, maintenance, and high reliability. The system collects parking data and transmits it to service providers, allowing drivers to obtain real-time information about available parking spaces and facilitating traffic congestion reduction. The working technique of the system involves users registering through a mobile application, checking slot availability, booking a slot, and obtaining a QR code for access to the parking area. The system tracks parking time and sends notifications to users when their time exceeds. The conclusion suggests that the proposed system can effectively address parking issues, but future improvements could include incorporating multiple parking areas, GPS-based location finding, and expanding the system to various malls and complexes.

All the papers focus on utilizing NB-IoT as the primary communication technology for data transmission in their respective systems. NB-IoT is chosen for its low power consumption, wide coverage, and high connectivity, making it suitable for low-power IoT devices. The papers also emphasize the integration of sensors to collect relevant data in their respective application domains. The sensors capture data related to irrigation, air quality, cow health, manhole cover status, slope monitoring, and parking availability, among others. These works also highlight the use of NB-IoT for reliable and efficient data transmission. NB-IoT enables stable communication over long distances while minimizing power consumption, making it ideal for IoT applications. Most of the studies describe the overall system architecture in layers, including hardware and software components. They typically include perception layers (sensors), communication layers (NB-IoT modules), and control layers (microcontrollers) to facilitate data collection, transmission, and processing.

On the other hand, these studies cover wide variety of different applications such as sustainable farming irrigation, air quality monitoring, cow health monitoring, industrial/environmental monitoring, smart street lighting, fire protection, manhole cover monitoring, slope disaster monitoring, and smart car parking. The specific domain determines the sensors, algorithms, and software design choices.

Also, the papers differ in the communication protocols and algorithms used. They explore and compare protocols such as TCP, UDP, CoAP, and MQTT to assess their suitability for low-power IoT devices. Additionally, some papers propose new algorithms for data clustering, warning index acquisition, and intelligent control strategies.

While there are similarities in terms of microcontrollers and communication modules used, each paper employs specific hardware and software components tailored to their application domain. These include different microprocessor units, gas sensors, accelerometers, STM32, BC-95, and ADXL345, among others.

The papers also vary in terms of system implementation scenarios, including field tests, real-time monitoring, cloud platform integration, and mobile application development. The focus is on verifying the feasibility, stability, and reliability of the systems in their respective contexts.

The cited works collectively demonstrate the versatility and potential of NB-IoT technology in various domains. They emphasize the benefits of NB-IoT, such as low power consumption, wide coverage, and reliable data transmission. While the papers share common properties such as the use of NB-IoT and sensor integration, they differ in application domains, protocols, hardware/software components, and system implementation. These variations highlight the adaptability of NB-IoT technology in addressing specific challenges and optimizing various IoT applications.

Even though there is a considerable amount of variation in the application domains and protocols in these previous works, there are no representation of the usage of NB-IoT paired with MQTT protocol running telemetry collection on the network itself that the user equipment is connected to.

4 DESIGN AND IMPLEMENTATION

This section presents the design and implementation details of our solution, also highlighting some of the choices and considerations made during the development process. First, the details about the finalized system will be presented. Following that, the implementation process would be detailed.

4.1 System Architecture

The basic system environment consists of the Cellular Internet of Things (C-IoT) transceiver, the microprocessor/computer, the MQTT broker and the cellular network that is being monitored.

The transceiver connects to the network, collects telemetry, and publishes the processed data from the computer to the MQTT broker through the same network. The computer simply operates the transceiver and processes the data provided by the transceiver. MQTT broker on the other hand, help visualizing and storing the telemetry data while providing essential MQTT functionality. Figure 5 represents a simplified visualization of the architecture.

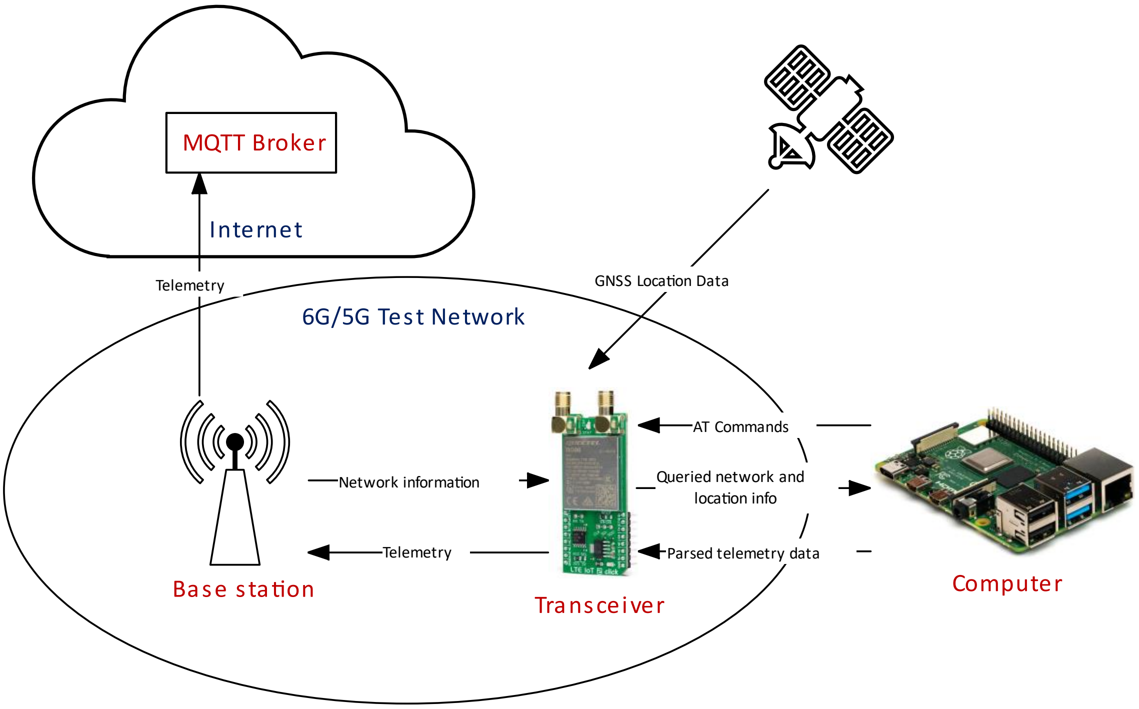


Figure 5. A simplified architecture schematic for the described system.

The hardware is managed by a C program that is ran by the computer and essentially control the transceiver, process the telemetry and publish the data to the MQTT broker.

4.2 The Transceiver

For our implementation, we wanted to see the feasibility of a system employing LTE Cat-NB1 technology, specifically the Quectel BG96 LTE Cat M1/NB1 module. For the development, we have used the MikroElektronika LTE IoT 2 Click breakout board [35] that consists of this module, which can be seen in Figure 6.

Figure 6. Picture of an "LTE IoT 2 Click" breakout board [35].

It is capable of employing GSM, LTE Cat-M1 and LTE Cat-NB1 technologies, and it comes with protocols such as TCP/IP, MQTT and LwM2M already embedded. The transceiver does not come with an integrated antenna. Instead, it comes with two separate SMA sockets for cellular and Global Navigation Satellite System (GNSS) application. For both cellular and GNSS connectivity, we are using external antennas as can be seen in Figure 7.

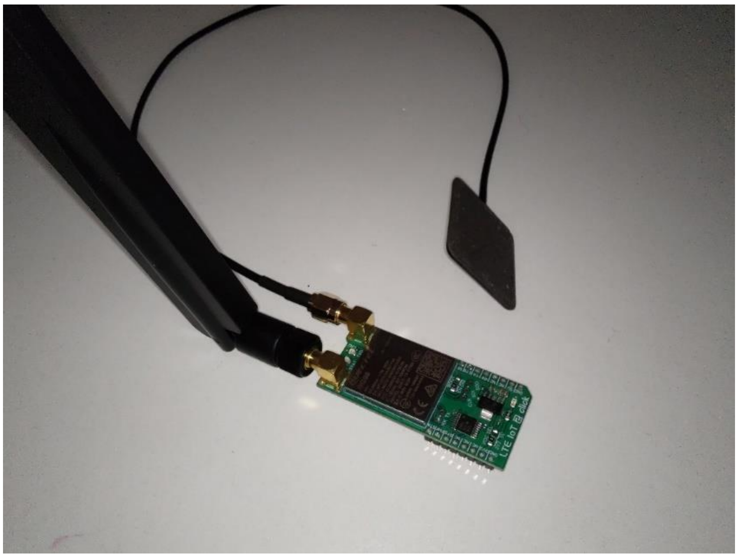


Figure 7. Transceiver with the both cellular and GNSS antennas connected.

The transceiver is controlled through its General Purpose Input/Output (GPIO) and Universal Asynchronous Receiver-Transmitter (UART) pins. Power state can be read and manipulated through its status and reset pins. Meanwhile rest of the functionality is achieved through the use of UART pins with the help of AT commands.

4.3 Computer

One of our aims was to integrate such transceiver with a microprocessor-based system or a computer. We have decided to base our system on a Raspberry Pi single board computer. Specifically, a Raspberry Pi 3 Model B. As can be seen in Figure 8, these single board computers are very compact.

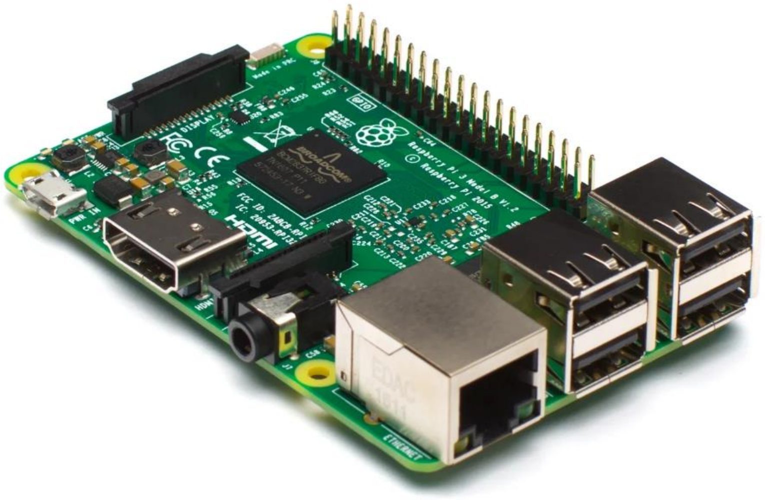


Figure 8. A Raspberry Pi 3 Model B [38].

Raspberry Pi is interfaced with the transceiver through its GPIO interface and also supplies power to the breakout board through its 5V and 3.3V supply pins.
The transceiver is powered by the supply pins of the Raspberry Pi device. The control and data transfer are done through GPIO and UART pins. The Figure 9 shows the wiring diagram of the pins between the transceiver and the Raspberry Pi computer GPIO pins and Figure 10 shows the assembled device including the connected GPIO pins.

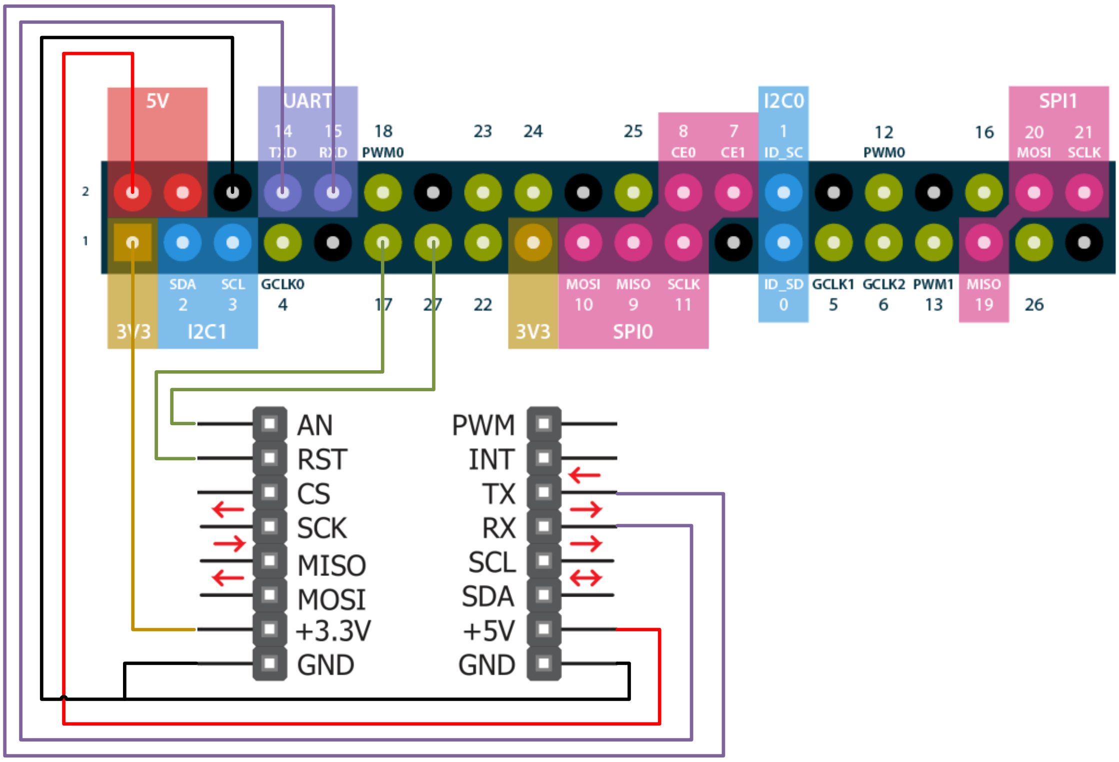


Figure 9. Pin wiring diagram between the transceiver pins and the Raspberry Pi GPIO pins.

The Linux system installed in the computer, which is a distribution specifically made for this hardware, already includes drivers for the GPIO pins compiled with the kernel. This makes the interfacing quite easy from the software aspect.

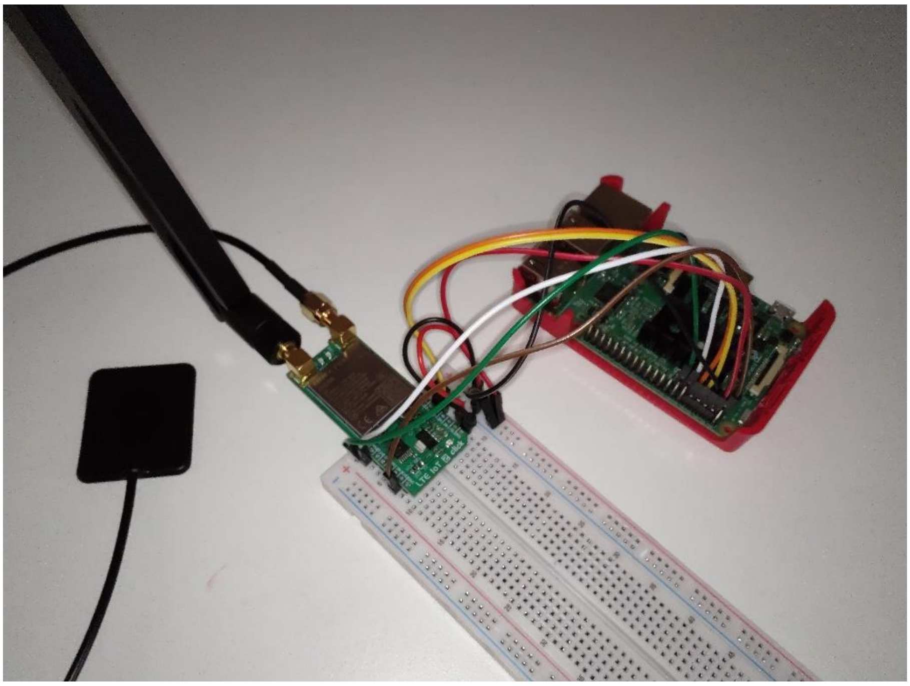


Figure 10. Transceiver and the computer assembled together.

4.4 Software

The hardware is managed by a program that was developed in C programming language, running in the Raspberry Pi computer. The program is called “network_monitor”. It is capable of controlling the transceiver through GPIO pins, conducting serial communication using the designated UART pins with AT commands, and parsing and processing the telemetry data provided by the transceiver. It can be found the address “*https://github.com/kutayboz/network-monitor*” by the time of the publication of this work.

The program is not implemented as a daemon, which means it does not run for a prolonged time in the background. It is called to execute one of its main functionalities and exits after completing its task. The program has 3 main functionalities:

- Powering the transceiver on and initializing the cellular and GNSS functionality
- Collecting telemetry data and publishing it to a MQTT broker
- Disconnecting from the network and powering the transceiver off

The initialization of the transceiver is achieved by calling the program with the “--init” argument. This functionality consists of the following actions:

1. Powering the transceiver on
2. Configuring the cellular and GNSS properties such as technology, operator name, etc.
3. Enabling the cellular and GNSS functionality, and establishing connection with the cellular network

After the initialization is completed, the program exits, and the transceiver sits idle while connected and registered to the network. Following this, the program can be called again periodically to publish telemetry to the MQTT broker. It is done by calling the program with the argument “--netinfo-pub”. This results in the following actions:

1. Retrieving the network telemetry by querying each property one by one
2. Parsing/formatting the telemetry data
3. Publishing the processed data to the MQTT broker

The collected telemetry data consists of:

- Operator name
- Network name
- Registration status (roaming, not roaming)
- Tracking area code
- Cell ID number
- Technology (LTE Cat-NB1, LTE Cat-M1, GSM)
- Band
- Channel number
- Network time
- GNSS location
- Signal quality measurements
    - Received Signal Strength Indicator (RSSI)
    - Reference Signal Received Power (RSRP)
    - Signal-to-Interference-plus-Noise Ratio (SINR)
    - Reference Signal Received Quality (RSRQ)
    - Bit Error Rate (BER)

The numerical signal quality measurement data are published into their own respective topics while the GNSS data is formatted into a JSON data conforming to the Adafruit IO Application Programming Interface (API) [39]. Rest of the data is text, and they are formatted in a way that makes them easy to print out.

Last main functionality would be the power off routine. This is done by calling the program with the argument “--stop". This results in:

1. Stopping the GNSS receiver
2. Disconnecting from the network
3. Powering the transceiver off

With these 3 main functions, the same program can be used for a simple telemetry publishment workflow, which is presented in Figure 11.

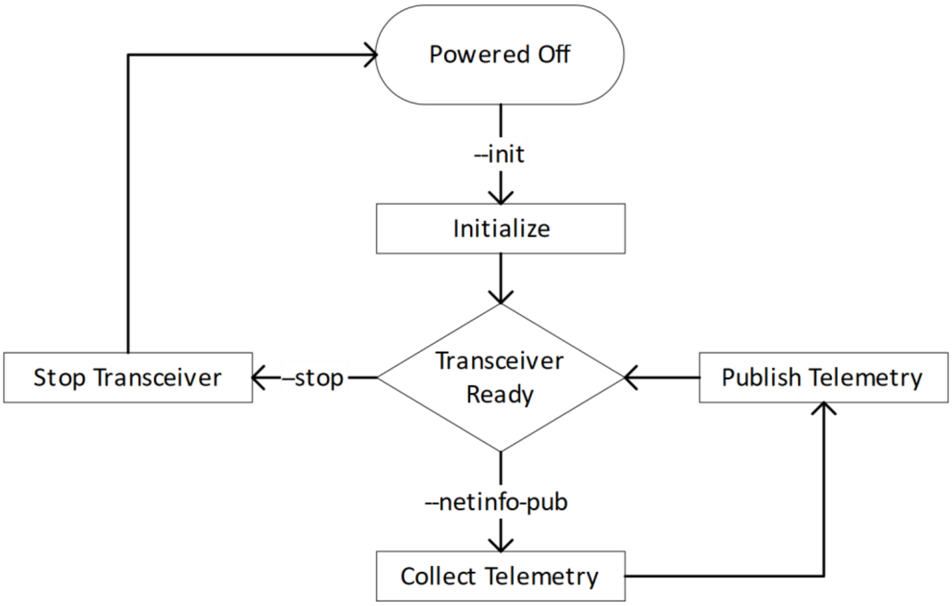
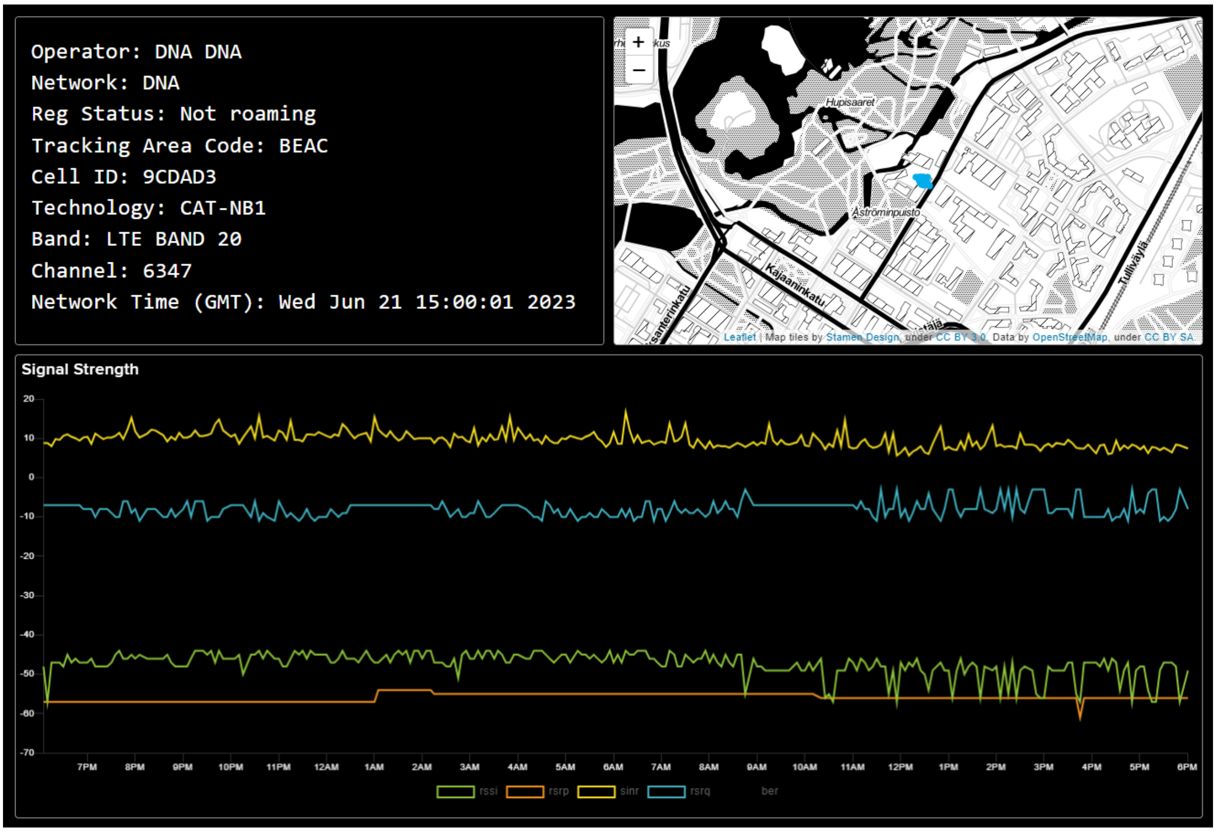


Figure 11. The flowchart of the intended routine for the program.

4.5 MQTT Broker/Server

As the MQTT broker, Adafruit IO was used. The main reason is, on top of providing the essential MQTT broker functionality, it is able to visualize the data in its web interface. This way it was possible to graph the signal quality measurements, map the GNSS location data and visualize other telemetry data. Figure 12 shows the mentioned web interface showing the all types of telemetry data that is being collected during a test run.



```
Operator: DNA DNA
Network: DNA
Reg Status: Not roaming
Tracking Area Code: BEAC
Cell ID: 9CDAD3
Technology: CAT-NB1
Band: LTE BAND 20
Channel: 6347
Network Time (GMT): Wed Jun 21 15:00:01 2023
```

Figure 12. Web interface of the MQTT broker with the collected telemetry from a test run visible.

The API of the broker provides a way to plot the GNSS location directly in the web interface as long as the data is provided in a certain JSON format. Other numeric data are published in their respective topics, and they are graphed on a single line graph. Rest of the data which are text only are printed directly.

The software used to publish the data allows any kind of broker to be used. If the user prefers, they can use any other broker, but they will need to find another way to visualize the data if they desire so.

4.6 Implementation

The implementation started with the initial attempts at interfacing the transceiver board with the Raspberry Pi computer. As the breakout board already included required circuits and components for the pin connections, such as voltage regulator circuits, circuit building consisted only the direct connection of the jumper cables as in Figure 9.

After setting the Raspberry Pi computer with the most up-to-date operating system, first serial communication with the transceiver was done manually through directly writing AT commands to the UART pins of the device.

Following this, a Subscriber Identification Module (SIM) card was obtained from a local service provider, DNA Oy., for the purpose of verifying the functionality of the transceiver in a commercial public network. This local service provider had very good LTE base station coverage that would provide a robust development environment, as can be seen in Figure 13.

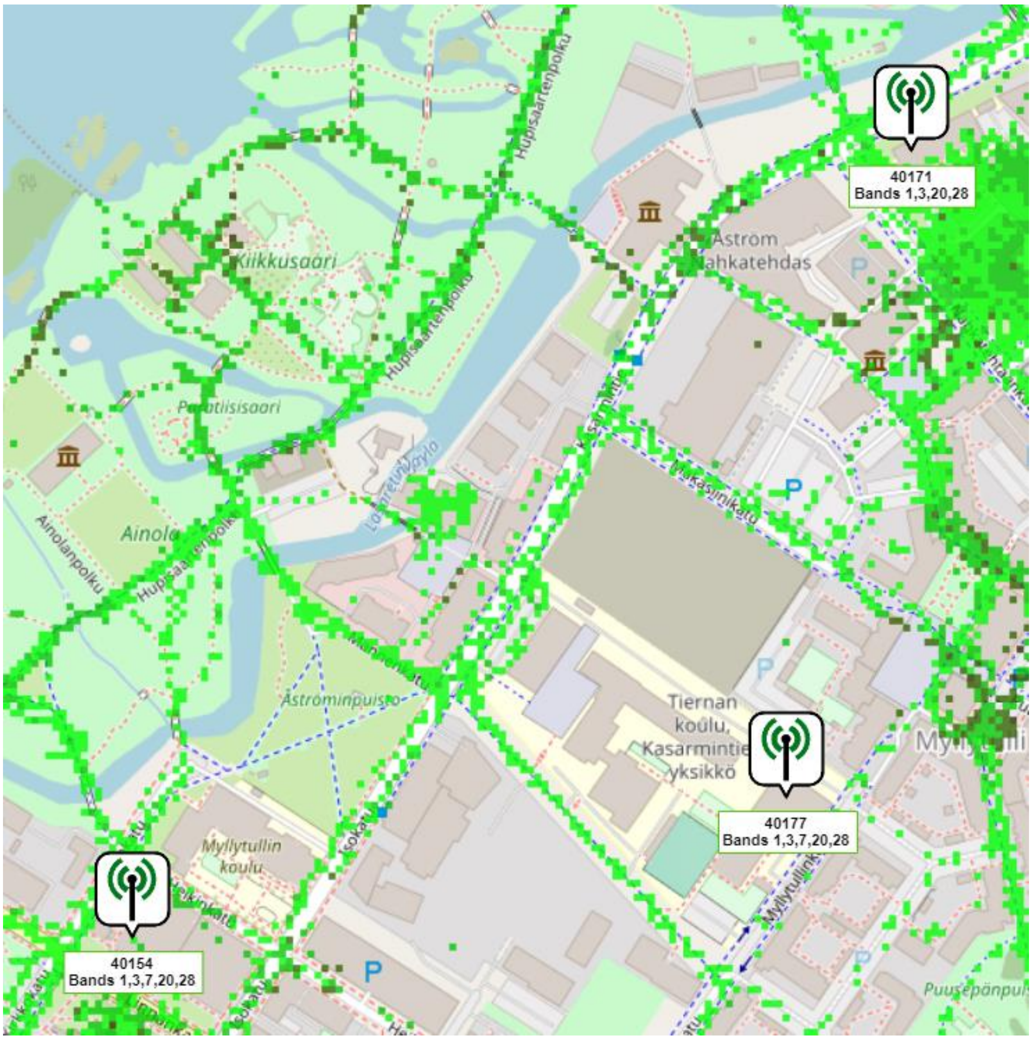
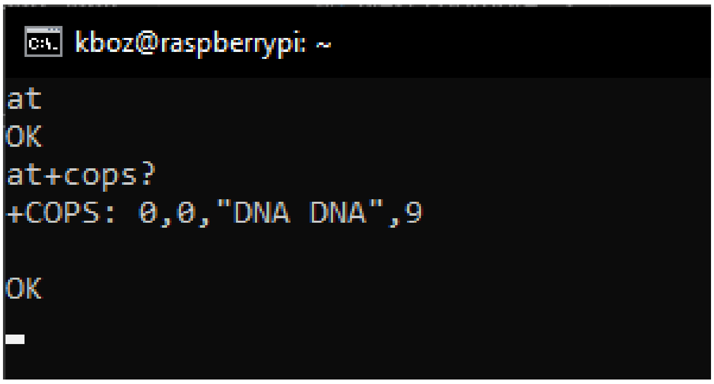


Figure 13. LTE base stations of DNA Oy. and the active bands in the vicinity of the public network test location [40].

By manually communicating with the device using AT commands through UART serial pins, the SIM card was successfully activated. The transceiver was forced into LTE Cat-NB1 mode and was commanded to automatically register to a nearby network.



```
at
OK
at+cops?
+COPS: 0,0,"DNA DNA",9

OK
```

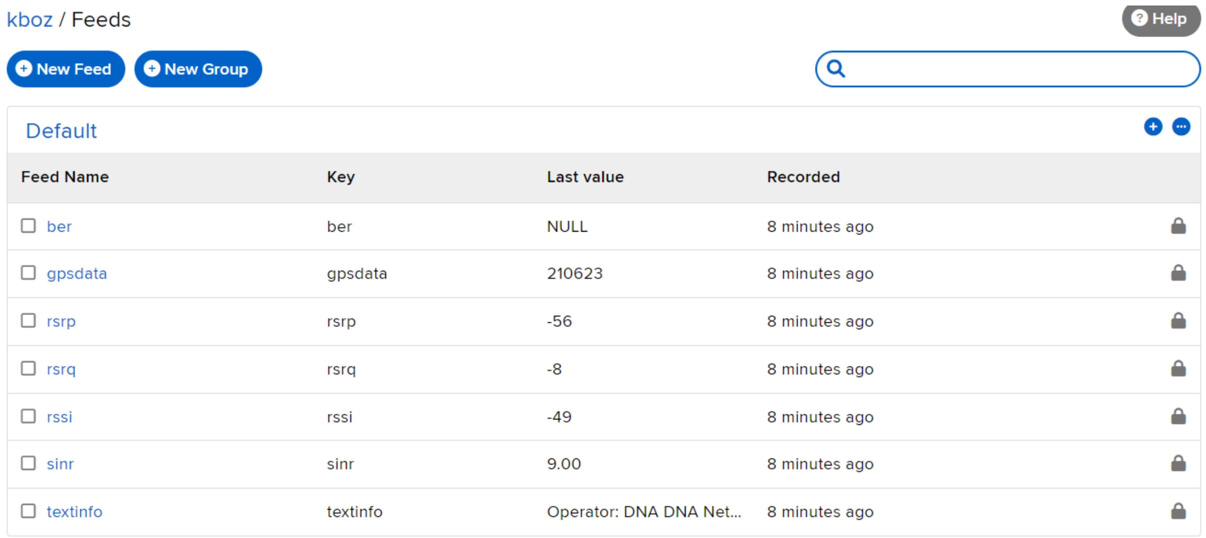
Figure 14. Transceiver returning the registered operator information.

Figure 14 shows the response of the transceiver. According to [41], the response in the figure describes that the device is in automatic operator selection mode and connected to the “DNA” operator using LTE Cat NB1 technology.

Following the successful registration with the network, the internet connectivity was tested by first sending pings to well known addresses such as Google and University of Oulu website. Small text-based data was also downloaded to verify it further.

After the bring-up of transceiver, the initial version of the control software was developed. This initial version was simply a quick way to turn the transceiver on and send a predetermined list of AT commands followed by the power off command. With the help of it, the required commands for the telemetry data and their responses were tested.

Before the implementation of the telemetry publish functionality, a broker from the Adafruit IO service was created. The MQTT functionality was tested by sending arbitrary data that is input manually.



Feed Name	Key	Last value	Recorded
☐ ber	ber	NULL	8 minutes ago
☐ gpsdata	gpsdata	210623	8 minutes ago
☐ rsrp	rsrp	-56	8 minutes ago
☐ rsrq	rsrq	-8	8 minutes ago
☐ rssi	rssi	-49	8 minutes ago
☐ sinr	sinr	9.00	8 minutes ago
☐ textinfo	textinfo	Operator: DNA DNA Net...	8 minutes ago

Figure 15. Overview of the topics and the data from the broker web interface.

After verifying that the transceiver was able to successfully publish data to a MQTT broker, which can be seen in Figure 15, the software was further developed. The end result was the current version which can query, parse and publish the telemetry data to the MQTT broker. The functionality of the was verified by plotting and comparing the received data in the MQTT broker web interface.

Final step was to register the system to the 6G/5G test network in the campus and test the whole functionality there. Adding the extra AT commands that configure the transceiver for the test network to the program and using the SIM card provided by the test network management, the system was successfully able to register, collect telemetry and publish data to the MQTT broker without any further intervention.

5 DISCUSSION

In this section, we will discuss the findings and implications of our solution as well as address some limitations and potential areas for future improvement.

5.1 System Performance

The implemented system demonstrated its capability to connect to the cellular network, collect telemetry data, and publish it to the MQTT broker successfully. The integration of the Quectel BG96 LTE Cat M1/NB1 module with the Raspberry Pi proved to be feasible and effective. The system was able to register to the network, retrieve network telemetry, and publish it to the MQTT broker without any major issues.

The MQTT broker, Adafruit IO, provided a user-friendly web interface for visualizing the telemetry data. The ability to graph the signal quality measurements, map the GNSS location data, and visualize other telemetry data proved to be valuable in monitoring and analyzing the cellular network performance.

5.2 Limitations

Despite the successful implementation, there are certain limitations to consider. First, the system was tested in a specific environment using the Quectel BG96 module and Raspberry Pi 3 Model B. While these components performed well in our setup, it may not be directly applicable to all scenarios. Different modules or single-board computers may have varying capabilities and requirements.

Furthermore, the current implementation focused on the collection and publication of telemetry data. Additional functionalities, such as data analysis, advanced processing, or integration with other systems, were not explored in depth. Future research could investigate these aspects to enhance the system's capabilities and provide more comprehensive insights into the cellular network.

5.3 Future Improvements

There are several potential areas for future improvement and expansion of the system. Firstly, incorporating advanced data analytics and machine learning algorithms can enable more sophisticated analysis of the collected telemetry data. This could provide valuable insights into network performance trends, anomalies, and predictive maintenance.

Moreover, the system can be enhanced to support real-time monitoring and alerts. By implementing event-driven mechanisms, such as triggering actions based on specific telemetry thresholds or network events, operators can be immediately notified of critical situations and take prompt actions to mitigate them.

Additionally, scalability and interoperability are essential factors to consider for a broader deployment. The system could be extended to support multiple transceivers and integrate with existing network management systems. This would enable a comprehensive monitoring and management solution for large-scale cellular networks.

Finally, considering the emergence of 6G and future generations of cellular networks, further research can be conducted to adapt the system to evolving technologies. Exploring the potential integration of emerging technologies, such as edge computing, network slicing, or dynamic spectrum sharing, can provide valuable insights into the performance and optimization of next-generation networks.

In conclusion, the designed and implemented system successfully demonstrated its capability to monitor and collect telemetry data from a cellular network using an LTE Cat-NB1 transceiver and a Raspberry Pi. The integration of these components, along with the MQTT broker, provided a practical solution for network monitoring and analysis.

While limitations exist, such as the specific hardware configuration and the focus on telemetry collection, the system presents a foundation for further research and enhancements. By incorporating advanced analytics, real-time monitoring, and scalability considerations, the system can evolve into a powerful tool for cellular network management.

The outcomes of this study contribute to the field of IoT and network management, offering insights into the design and implementation of monitoring systems for cellular networks. The knowledge gained from this research can guide future developments and improvements in the area of cellular network monitoring and optimization.

6 SUMMARY

The integration of cellular Internet of Things (IoT) devices into modern networks has become a topic of great interest, offering the potential for diverse applications and improved network monitoring capabilities. This thesis aims to explore the integration and deployment of an aftermarket cellular IoT transceiver on a 6G/5G test network, specifically focusing on the evaluation of its performance as a network telemetry provider.

The integration process involves utilizing a NB-IoT transceiver paired with a Raspberry Pi microprocessor to collect network telemetry and establish connectivity via the MQTT protocol. The system is initially tested in a public cellular network, where successful connection, internet connectivity, and both MQTT and TCP/IP establishment are achieved.

To monitor and gather essential network information, a network monitoring utility is developed. This utility collects data such as network identifiers, module registration status, band/channel details, signal strength measurements, and GPS positions. The collected data is then published to an MQTT broker, with the Adafruit IO platform serving as the chosen broker that also provides a user interface for data visualization.

Furthermore, the system is configured and deployed on a 6G/5G test network at the University of Oulu. The functionality developed and tested in the public network is successfully maintained, enabling continuous monitoring and analysis of network data in the test network environment.

The findings highlight the feasibility of utilizing such systems for network monitoring purposes and showcase the potential for IoT applications within cellular networks.

The thesis objectives can be summarized as follows:

1. Integrate an aftermarket cellular IoT transceiver into a working microprocessor system.
2. Establish reliable connectivity to the designated 6G/5G test network.
3. Develop a network monitoring utility that employs MQTT to publish collected data.
4. Visualize the collected telemetry data.

7 REFERENCES

[1] K. Kaur, “A Survey on Internet of Things – Architecture, Applications, and Future Trends,” in 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India: IEEE, Dec. 2018, pp. 581–583. doi: 10.1109/ICSCCC.2018.8703341.

[2] P. V. Paul and R. Saraswathi, “The Internet of Things — A comprehensive survey,” in 2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), Melmaruvathur: IEEE, Mar. 2017, pp. 421–426. doi: 10.1109/ICCPEIC.2017.8290405.

[3] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications,” IEEE Internet Things J., vol. 4, no. 5, pp. 1125–1142, Oct. 2017, doi: 10.1109/JIOT.2017.2683200.

[4] B. E. Benhiba, A. A. Madi, and A. Addaim, “Comparative Study of The Various new Cellular IoT Technologies,” in 2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra: IEEE, Dec. 2018, pp. 1–4. doi: 10.1109/ICECOCS.2018.8610508.

[5] M. Agiwal, A. Roy, and N. Saxena, “Next Generation 5G Wireless Networks: A Comprehensive Survey,” IEEE Commun. Surv. Tutorials, vol. 18, no. 3, pp. 1617–1655, 2016, doi: 10.1109/COMST.2016.2532458.

[6] “5GTN - 5G Test Network.” https://5gtn.fi/

[7] “LPWA BG96 Cat M1/NB1/EGPRS,” Quectel. https://www.quectel.com/product/lpwa-bg96-cat-m1-nb1-egprs (accessed Jun. 18, 2023).

[8] “Adafruit IO.” https://io.adafruit.com (accessed Jun. 18, 2023).

[9] L. M. Dang, Md. J. Piran, D. Han, K. Min, and H. Moon, “A Survey on Internet of Things and Cloud Computing for Healthcare,” Electronics, vol. 8, no. 7, p. 768, Jul. 2019, doi: 10.3390/electronics8070768.

[10] “Recommendation ITU-T Y.2060: Overview of the Internet of things.” International Telecommunication Union, 2012.

[11] “Mobile IoT Deployment Map,” Internet of Things. https://www.gsma.com/iot/deployment-map/ (accessed Jun. 20, 2023).

[12] R. Boisguene, S.-C. Tseng, C.-W. Huang, and P. Lin, “A survey on NB-IoT downlink scheduling: Issues and potential solutions,” in 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Jun. 2017, pp. 547–551. doi: 10.1109/IWCMC.2017.7986344.

[13] Y.-P. E. Wang et al., “A Primer on 3GPP Narrowband Internet of Things,” IEEE Commun. Mag., vol. 55, no. 3, pp. 117–123, Mar. 2017, doi: 10.1109/MCOM.2017.1600510CM.

[14] “3GPP TS 36.214 Rel.17: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements.” The 3rd Generation Partnership Project (3GPP).

[15] “3GPP TS 36.211 Rel.17: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation.” The 3rd Generation Partnership Project (3GPP).

[16] N. Naik, “Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP,” in 2017 IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria: IEEE, Oct. 2017, pp. 1–7. doi: 10.1109/SysEng.2017.8088251.

[17] “MQTT Essentials - All Core Concepts explained.” https://www.hivemq.com/mqtt-essentials/ (accessed Jun. 20, 2023).

[18] “Satelliittipaikannus,” Maanmittauslaitos. https://www.maanmittauslaitos.fi/tutkimus/teematietoa/satelliittipaikannus (accessed Jun. 24, 2023).

[19] “JSON.” https://www.json.org/json-en.html (accessed Jun. 25, 2023).

[20] “What is Quality of Service (QoS) in Networking?,” Fortinet. https://www.fortinet.com/resources/cyberglossary/qos-quality-of-service (accessed Jun. 25, 2023).

[21] “UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter | Analog Devices.” https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html (accessed Jun. 25, 2023).

[22] R. & S. G. & C. KG, “Understanding UART.” https://www.rohde-schwarz.com/fi/products/test-and-measurement/essentials-test-equipment/digital-oscilloscopes/understanding-uart_254524.html (accessed Jun. 25, 2023).

[23] J. Cardoso, A. Gloria, and P. Sebastiao, “A Methodology for Sustainable Farming Irrigation using WSN, NB-IoT and Machine Learning,” in 2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Corfu, Greece: IEEE, Sep. 2020, pp. 1–6. doi: 10.1109/SEEDA-CECNSM49515.2020.9221791.

[24] T. Manninen, “Design and Implementation of Energy Efficient Narrowband IoT Sensor Platform,” Aalto University, Espoo, 2020.

[25] M. Stusek, K. Zeman, P. Masek, J. Sedova, and J. Hosek, “IoT Protocols for Low-power Massive IoT: A Communication Perspective,” in 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and

Workshops (ICUMT), Dublin, Ireland: IEEE, Oct. 2019, pp. 1–7. doi: 10.1109/ICUMT48472.2019.8970868.

[26] Y. Cheng, X. Xu, Y. Du, P. Guan, S. Liu, and L. Zhao, “Design of Air Quality Monitoring System Based on NB-IoT,” in 2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China: IEEE, Jul. 2019, pp. 385–388. doi: 10.1109/ICPICS47731.2019.8942533.

[27] B. Khan and C. Pirak, “Experimental Performance Analysis of MQTT and CoAP Protocol Usage for NB-IoT Smart Meter,” in 2021 9th International Electrical Engineering Congress (iEECON), Pattaya, Thailand: IEEE, Mar. 2021, pp. 65–68. doi: 10.1109/iEECON51072.2021.9440273.

[28] P. Chen, “Dairy Cow Health Monitoring System Based on NB-IoT Communication,” in 2019 International Conference on Electronic Engineering and Informatics (EEI), Nanjing, China: IEEE, Nov. 2019, pp. 393–396. doi: 10.1109/EEI48997.2019.00091.

[29] M. Klymash, T. Maksymyuk, S. Dumych, and O. Yaremko, “Designing the Industrial and Environmental Monitoring System based on the Internet of Things Architecture,” in 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), Lviv: IEEE, Sep. 2018, pp. 187–190. doi: 10.1109/IDAACS-SWS.2018.8525545.

[30] S. Chen, G. Xiong, J. Xu, S. Han, F.-Y. Wang, and K. Wang, “The Smart Street Lighting System Based on NB-IoT,” in 2018 Chinese Automation Congress (CAC), Xi’an, China: IEEE, Nov. 2018, pp. 1196–1200. doi: 10.1109/CAC.2018.8623281.

[31] T. Li and P. Hou, “Application of NB-IoT in Intelligent Fire Protection System,” in 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Jishou, China: IEEE, Sep. 2019, pp. 203–206. doi: 10.1109/ICVRIS.2019.00057.

[32] X. Guo, B. Liu, and L. Wang, “Design and Implementation of Intelligent Manhole Cover Monitoring System Based on NB-IoT,” in 2019 International Conference on Robots & Intelligent System (ICRIS), Haikou, China: IEEE, Jun. 2019, pp. 207–210. doi: 10.1109/ICRIS.2019.00061.

[33] M. Xie, Z. He, L. Liang, Z. Wu, and G. Sun, “Slope disaster monitoring and early warning system based on 3D-MEMS and NB-IoT,” in 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China: IEEE, Dec. 2019, pp. 90–94. doi: 10.1109/IAEAC47372.2019.8997685.

[34] M. Praveen and V. Harini, “NB-IOT based smart car parking system,” in 2019 International Conference on Smart Structures and Systems (ICSSS), Chennai, India: IEEE, Mar. 2019, pp. 1–5. doi: 10.1109/ICSSS.2019.8882847.

[35] “LTE IoT 2 click - IoT solution | MikroElektronika,” MIKROE. http://www.mikroe.com/lte-iot-2-click (accessed Jun. 19, 2023).

[36] "network_monitor Software Repository", Boz K. https://github.com/kutayboz/network-monitor (accessed Jun. 24, 2023).

[37] https://github.com/kutayboz/network-monitor

[38] "Raspberry Pi 3 Model B — Raspberry Pi Australia," Pi Australia. https://raspberry.piaustralia.com.au/products/raspberry-pi-3-model-b (accessed Jun. 21, 2023).

[39] "Adafruit IO API Reference." https://io.adafruit.com/api/docs/mqtt.html (accessed Jun. 20, 2023).

[40] CellMapper, "Cellular Tower and Signal Map," CellMapper. https://www.cellmapper.net/map?MCC=-1&MNC=-1, https://www.cellmapper.net/map (accessed Jun. 21, 2023).

[41] "BG96 AT Commands Manual." Quectel Wireless Solutions Co., Ltd., Jan. 25, 2019.