

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Stanley Ohumegbulem Osajeh 196388IVEM

CAT-M1 COVERAGE ANALYSIS FOR DEMO CAMPUS ENVIRONMENT

Master's thesis

Supervisors: Muhammad Mahtab
Alam
Professor

Sven Päränd
PhD

Tallinn 2021

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Stanley Ohumegbulem Osajeh 196388IVEM

CAT-M1 KATVUSE ANALÜÜS DEMOLINNAKU KESKKONNAS

Magistritöö

Juhendaja: Muhammad Mahtab
Alam
Professor

Sven Päränd
PhD

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Stanley Ohumegbulem Osajeh

10.05.2021

Abstract

The steady development of Low Power Wide Area Network (LPWAN) technologies over the years has resulted in two broad categories of this technology: cellular and non-cellular LPWAN technologies. While both have their merits and demerits, cellular LPWAN technologies present a unique advantage of increased global penetration by leveraging the global coverage of the LTE technology. Furthermore, the forecasted proliferation of IoT devices and connections in the coming years puts a crucial demand on IoT network service providers to frequently evaluate their network's performance to ensure a robust network that can support this envisaged scale and provide quality service to users. This crucial demand is the motivation for this thesis work as it would help give service providers some insights into their network service delivery from the users' perspective.

There are several performance evaluation methods highlighted in this thesis, but the coverage analysis method is the performance evaluation method used as it considers the signal resources between the User Equipment (UE) and the evolved Node B (eNB). This thesis work approaches this task by first developing an end-to-end IoT system and subsequently conducting a measurement campaign to collect the required data for analysis. The measurement campaign was conducted on a specifically selected building with four floors (elevation levels) within TalTech's main campus building. Conducting measurements on various elevation levels also helped investigate the possible impact of altitude on the coverage quality. The data collected is then analysed, at both device and floor levels, to understand the possible experience of users on the network. The insight obtained from the data collected indicates that users on all floors in the measured coverage area within TalTech main building would generally experience quality service.

This thesis is written in English and is 73 pages long, including 6 chapters, 36 figures and 10 tables.

Annotatsioon

CAT-M1 katvuse analüüs demolinnaku keskkonnas

Madala võimsusega laivõrkude (inglise keeles *low power wide area networks*, LPWAN) järjepidev areng aastate jooksul on toimunud kahes suunas: kärgevõrkude (inglise keele *cellular*) ning mittekargevõrkude (inglise keeles *non cellular*) tehnoloogiad. Kuigi mõlemal on oma positiivsed küljed ning ka puudujäägid, siis kärgevõrkudel põhinevatel tehnoloogiatel on suurem võimalus saavutada ülemaailmselt laiemat levikut tänu sellele, et need põhinevad neljanda põlvkonna mobiilsidevõrkudel (inglise keeles *long term evolution*, LTE). Lisaks, eeldatav asjade interneti (inglise keeles *internet of things*, IoT) seadmete arvu kasv lähiaastatel paneb IoT võrguteenuse pakkujad olukorda, kus nad peavad oma võrkude võimekust järjepidevalt hindama, tagamaks kasutajatele parim teenuse kvaliteet. Viimasena mainitu ongi käesoleva lõputöö aluseks – eesmärk on, et teenusepakkujad suudaksid hinnata teenuse kvaliteeti lõppkasutaja seisukohast.

Kuigi lõputöö käsitleb mitut võrgu võimekuse hindamise meetodit, keskendub see siiski raadiolevi katvusanalüüsile kuna selle abil on võimalik saada ülevaade reaalistest raadioressurssidest lõppkasutaja seadme ja tugijaama (inglise keeles *evolved node B*, eNB) vahel. Töö lõpptulemus saavutatakse esmalt läbi IoT süsteemi väljatöötamise ja rakendamise, mõõtmiste läbiviimise ning saadud tulemuste analüüsi. Raadiovõrgu mõõtmised teostatakse Tallinna Tehnikaülikooli linnakus, selleks eraldi välja valitud hoones, neljal korrusel. Erinevatel korrustel mõõtmiste tegemise eesmärgiks on uurida kõrguse mõju katvusele ning seeläbi teenuse kvaliteedile. Kogutud tulemused ning nende põhjal teostatud analüüs näitab, et valitud hoone kõigil korrustel kogeksid lõppkasutajad kvaliteediprobleeme.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 73 leheküljel, 6 peatükki, 36 joonist ja 10 tabelit.

List of abbreviations and terms

3GPP	Third Generation Partnership Project
4G	Fourth Generation
ADC	Analog to Digital Converter
ACK	Acknowledgement
APN	Access Point Name
AT	Attention
AWS	Amazon Web Services
BPSK	Binary Phase-Shift Keying
CAN	Controller Area Network
CPU	Central Processing Unit
CSS	Chirp Spread Spectrum
DAC	Digital to Analog Converter
DC	Direct Current
eDRX	Extended Discontinuous Reception
eNB	Evolved Node B
EC2	Elastic Compute Cloud
FIN	Finish
FTP	File Transfer Protocol
GMT	Greenwich Mean Time
GND	Ground
GSMA	Global System of Mobile Communications Association
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IO	Input/Output
IoT	Internet of Things
IP	Internet Protocol
IR	Infrared
KPI	Key Performance Indicator
LED	Light Emitting Diodes
LoRa	Long Range
LPWAN	Low Power Wide Area Network

LTE-M	Long Term Evolution for Machine Type Communication
M2M	Machine-To-Machine
MAR	Mobile Autonomous Reporting
MB	MegaByte
MCL	Maximum Coupling Loss
MCU	Microcontroller Unit
M-PBCH	LTE-M Physical Broadcast Channel
M-PDSCH	LTE-M — Physical Downlink Shared Broadcast Channel
M-PDCCH	LTE-M Physical Downlink Control Channel
M-PRACH	LTE-M Physical Random Access Channel
M-PSCH	LTE-M Physical Synchronization Channel
M-PUSCH	LTE-M Physical Uplink Control Channel
M-PUCCH	LTE-M Physical Uplink Shared Channel
MQTT	Message Queuing Telemetry Transport
NACK	Negative Acknowledgement
NB	Narrowband
OFDMA	Orthogonal Frequency Division Multiple Access
PAR	Positive Acknowledgement and Retransmission
PBCH	Physical Broadcast Channel
Pmod	Peripheral Module
PRB	Physical Resource Block
PSD	Power Spectral Density
PSH	Push
PSM	Power Saving Mode
PSS	Primary Synchronization Sequence
PWM	Pulse Width Modulation
QoS	Quality of Service
RGB	Red Green Blue
RISC	Reduced Instruction Set Computer
RSSI	Received Signal Strength Indicators
RST	Reset
SC-FDMA	Single-Carrier Frequency Division Multiple Access
SIM	Subscriber Identification Module
SINR	Signal to Interference plus Noise Ratio
SNR	Signal to Noise Ratio
SRAM	Static Random Access Memory
SCP	Secure Copy
SPI	Serial Peripheral Interface
SSH	Secure Shell

SSS	Secondary Synchronization Sequence
SYN	Synchronize
TalTech	Tallinn University of Technology
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver Transmitter
UE	User Equipment
USB	Universal Serial Bus
URG	Urgent
VCC	Virtual Common Controller
WH	Watt Hours

Table of contents

1 Introduction	14
1.1 Background.....	15
1.2 Purpose	17
1.3 Overview of the Thesis.....	18
2 State of the Art.....	20
2.1 LTE Cat-M1 Technology Overview.....	20
2.2 LTE Cat-M1 Radio Resource	21
2.2.1 Downlink Radio Resource.....	22
2.2.2 Downlink Physical Channels.....	23
2.2.3 Uplink Radio Resource.....	24
2.2.4 Uplink Physical Channels.....	24
2.3 Performance Evaluation Methods	25
2.4 Related Work.....	26
3 Methodology.....	28
3.1 System Architecture	28
3.1.1 UE Node	29
3.1.2 Cloud Node.....	34
3.1.3 Communication Protocols	36
3.2 System Development Tools.....	41
3.3 System Design and Deployment.....	43
3.3.1 UE Node Design.....	44
3.3.2 Cloud Node Design	45
3.3.3 Deployment	46
4 Measurement Results.....	48
4.1 First Floor	49
4.2 Second Floor.....	50
4.3 Third Floor.....	52
4.4 Fourth Floor.....	54
5 Discussion.....	58

5.1 Results Analysis	58
5.2 Future Work.....	59
6 Summary.....	61
References	63
Appendix 1 – Non-exclusive Licence for Reproduction and Publication of a Graduation Thesis.....	68
Appendix 2 – MQTT Subscriber Python Script.....	69
Appendix 3 – UE Node Program Code	70

List of figures

Figure 1: Physical resource block for 1.4 MHz LTE frequency division duplexing [14]	22
Figure 2: Downlink time structure for LTE Cat-M1 radio access channel [10].....	23
Figure 3: Uplink time structure with 2.5 kHz subcarrier spacing for LTE Cat-M1 radio access channel [10].....	24
Figure 4: LTE Cat-M1 end-to-end system architecture	29
Figure 5: OpenMV Cam H7 device [26]	30
Figure 6: Avnet Silica NB-IoT Sensor Shield [31]	32
Figure 7: Schematic diagram of the UE node connections	33
Figure 8: Complete UE node	34
Figure 9: Schematic of the end-to-end system architecture	35
Figure 10: TCP/IP protocol stack [37]	37
Figure 11: TCP segment header structure [39].....	37
Figure 12: TCP three-way handshake for connection establishment [39].....	38
Figure 13: TCP connection termination [39].....	38
Figure 14: MQTT publish/subscribe architecture [42].....	39
Figure 15: UARTs connection for serial data transmission [44].....	41
Figure 16: UARTs connection with data bus	41
Figure 17: UE node program flow.....	44
Figure 18: MQTT subscriber program flow	45
Figure 19: Map view of devices installed.....	46
Figure 20: UE node installed on a floor.	47
Figure 21: Signal strength graph for each UE node device on the first floor during morning hours.....	49
Figure 22: Signal strength graph for the first floor during morning hours.....	49
Figure 23: Signal strength graph for each UE node device on the first floor during evening hours.....	49
Figure 24: Signal strength graph for the first floor during evening hours.....	50

Figure 25: Signal strength graph for each UE node device on the second floor during morning hours.....	51
Figure 26: Signal strength graph for the second floor during morning hours	51
Figure 27: Signal strength graph for each UE node device on the second floor during evening hours.....	51
Figure 28: Signal strength graph for the second floor during evening hours	52
Figure 29. Signal strength graph for each UE node device on the third floor during morning hours.....	53
Figure 30: Signal strength graph for the third floor during morning hours.....	53
Figure 31: Signal strength graph for each UE node device on the third floor during evening hours.....	53
Figure 32: Signal strength graph for the third floor during evening hours.....	54
Figure 33: Signal strength graph for each UE node device on the fourth floor during morning hours.....	55
Figure 34: Signal strength graph for the fourth floor during morning hours	55
Figure 35: Signal strength graph for each UE node device on the fourth floor during evening hours.....	55
Figure 36: Signal strength graph for the fourth floor during evening hours	56

List of tables

Table 1: Difference between cellular and non-cellular LPWAN technologies [7] [8]...	15
Table 2: Main differences between NB-IoT and LTE Cat-M1 [7].....	16
Table 3: Performance objectives for LTE Cat-M1 technology [10] [12].....	20
Table 4: OpenMV Cam H7 pinouts description [28]	30
Table 5: Avnet Silica NB-IoT Sensor Shield pinouts description [31]	33
Table 6: Types of MQTT control packet [43]	39
Table 7: Approximate distance and elevation of each device installation during campaign.....	46
Table 8: NB-IoT Signal Strength (RSSI) reference values [18].....	48
Table 9: Average signal strength values for each floor	57
Table 10: Total packet delivery for each floor	57

1 Introduction

Humans have communicated with themselves and their environment in different ways since existence. These ways have evolved over the years, with an origin traced as far back as approximately 500,000 BCE from the beginning of human speech through Writing, Printing technology, Telecommunications, Radio and Television, and now the Internet [1]. The emergence of telecommunications and internet technologies has revolutionized the way humans interact in recent times; for example, people can now promptly have effective video conversations with the aid of a handheld mobile device while in different geographic locations. These remarkable possibilities have not only applied to humans as we nowadays see the exchange of information among machines enabled by Machine-To-Machine Communications (M2M) and Internet of Things (IoT) technologies.

The IoT technology promises to unleash a plethora of opportunities and possibilities in human life. For example, enterprises are already leveraging IoT to create previously impossible use-cases such as asset tracking, augmented reality, and condition-based monitoring [2]. Also, several other applications of the IoT technology, such as wearables, smart homes, smart energy meters, that are focused on the consumers are in rapid development owing to the wide adoption of IoT and potential market opportunities. The Global System of Mobile Communications Association (GSMA) predicts a global market opportunity of \$1.1 trillion in revenue excluding IoT hardware revenue by 2025 and a total estimate of 25 billion IoT connections [3]. McKinsey & Company projects the number of IoT-connected devices globally to increase to 43 billion devices by 2023, which indicates an increase of approximately threefold from 2018. This growth in IoT-connected devices is projected to be influenced by reliable mobile connectivity, new sensors, and high computing power [4]. A significant type of communication technology that would help drive up to billions of new IoT-connected devices is the Low-Power Wide Area Network (LPWAN) technology [5].

LPWAN technology is a wireless narrowband technology designed to support massive IoT applications and M2M communication. These applications are generally required to

have a low cost, consume low power, have good performance at a long-range, use low data rates, and operate within a relatively small bandwidth. There are several types of LPWAN technology already deployed, and they can be broadly categorized into cellular and non-cellular LPWAN technologies [3]. Cellular LPWAN technologies, which includes Narrowband IoT (NB-IoT) and Long Term Evolution for Machines (LTE-M) (also called Category M1, Cat-M1 or LTE Cat-M1), was developed by the Third Generation Partnership Project (3GPP) to be deployed in the licensed spectrum and interoperate with existing mobile cellular communication networks like 4G; while non-cellular LPWAN technologies, such as Ingenu, Sigfox and Long Range (LoRa), operate in the non-licensed frequency bands outside of mobile cellular communication networks [6]. Although these two categories are suitable in various use-cases, the cellular LPWAN technology shows a better opportunity for wider deployment because of the global coverage of the already existing cellular networks.

1.1 Background

Cellular and non-cellular LPWAN technologies have different design models but target a similar market. Table 1 shows the general difference between both technologies. Among the non-cellular LPWAN technologies, Sigfox and LoRa are taking the lead [7]. Sigfox is an ultra-narrowband technology that uses Binary Phase-Shift Keying (BPSK) modulation and operates a network operator-style business model where the technology stack is a closed market managed by the company, also named Sigfox, and the Sigfox-enabled endpoints is an open market; while LoRa is a spread-spectrum technology that uses Chirp Spread Spectrum (CSS) modulation and runs a generally more open and flexible business model.

Table 1: Difference between cellular and non-cellular LPWAN technologies [7] [8]

Parameter	Cellular	Non-cellular
Band	Licensed cellular band	Unlicensed band
Power consumption	Less efficient	More energy efficient
Deployment coverage	Global	Limited

Data rates	Higher – up to 1Mbps	Varies for different standards but generally lesser throughput
Latency	Low	Relatively higher
Standardization	3GPP	Different governing bodies for different standards

On the other hand, there are three cellular LPWAN technologies, but the most popular are NB-IoT and Cat M1. While NB-IoT and LTE Cat-M1 utilize the same underlying technology, Cat M1 has a higher bandwidth, data rate, and it is more suitable for mission-critical real-time applications. Table 2 highlights the essential differences between NB-IoT and Cat M1.

Table 2: Main differences between NB-IoT and LTE Cat-M1 [7]

Parameter	LTE Cat-M1	NB-IoT
Bandwidth	1.4MHz	200kHz
Data rates	1Mbps	250kbps
Deployment	In-band LTE	Standalone, In-band LTE, Guard-band LTE
Use-case suitability	More suitable for mission-critical use-cases, mobility applications with real-time communications	More suitable for static sensor use-cases with limited mobility
Network coverage	156dB	164dB

The LTE Cat-M1 technology standard has been specified by the 3GPP from release 13 to 15 [9]. It operates within the LTE frequency band and has the following main performance objectives: to provide improved indoor coverage of a Maximum Coupling Loss (MCL) of up to 156dB, support a massive number of low throughput devices of about 52,547 devices within a cell sites sector in an urban area, achieve low device cost

through reduced device complexity, ensure improved power efficiency to achieve up to ten years battery life with a battery capacity of a maximum of 5 Watt-hour (Wh) in worst coverage case scenarios, and accommodate a higher latency of up to 10 seconds for application with fairly strict delay requirement [10]. LTE Cat-M1 achieves these performance objectives through its features, which includes: using Orthogonal Frequency Division Multiple Access (OFDMA) and Single-Carrier Frequency Division Multiple Access (SC-FDMA) for its downlink and uplink physical layer access technology, supporting user device power class 3 (23dBm), 5 (20dBm), and 6 (14dBm) with power-saving mode (PSM) and extended Discontinuous Reception (eDRX) for power efficiency, supporting full and half communication duplex modes, supporting repetition of up to 32 times (for coverage enhancement mode A) or 2048 times (for coverage enhancement mode B) for uplink and downlink data channel transmission [9].

1.2 Purpose

Achieving all the performance objectives of the LTE Cat-M1 technology is potentially challenging for network service providers. The support for a massive number of devices and the improved indoor coverage objectives of the LTE Cat-M1 technology is particularly strenuous, especially where network service providers have a high user density at areas that are poorly covered with the LTE network and must provide reasonable service quality to such users. Therefore, understanding the network service performance level within the coverage area of the service provider is imperative to provide quality service.

The Tallinn University of Technology (TalTech) campus has LTE Cat-M1 base stations already installed to deliver IoT network communication service to IoT devices within the campus. It is, therefore, imperative to predict the user experience of devices that would connect to this LTE Cat-M1 network within the campus. Analysis of the coverage performance of IoT network services within the campus coverage area can provide insight for such prediction. This insight would enable the service provider to determine where service optimization is required.

The need for network service providers to be able to understand their users' experience is even more important nowadays as the number of IoT use-cases continues to increase and the world transitions into a smarter global environment. For example, TalTech plans to

become a climate smart TalTech by 2035 [11]. This already suggests that it would need to install numerous IoT devices and sensors, which would have to communicate, at various locations within the campus, including potentially challenging areas like basements or underground parking lots. Understanding the service performance at different coverage areas would help to ensure that these connected IoT devices have the network resources they need to function adequately.

One way to gain this performance insight is by coverage analysis. The coverage analysis method scrutinizes the connection signal strength parameters of the connected IoT user device with respect to its location within the LTE Cat-M1 base station coverage area. This signal strength is considered when the base station determines the number of resources to allocate to the device for transmission, which affects the service performance. It is safe to say, on a general note, that the higher the signal strength, the better the quality of service.

This thesis, therefore, is primarily undertaken to understand the network service performance level of the LTE Cat-M1 network service installed in TalTech campus for IoT devices within an area in the campus by collecting and analysing the signal strength data. This insight can help service providers determine if network service optimization is required and provide a better experience to their users. It is noteworthy that, although TalTech campus is the reference location used for this work, the system developed is applicable in other environments with LTE Cat-M1 network service installed. The goal of this thesis work is achieved by designing an end-to-end IoT system that collects real-life image data and sends it, along with its network-related data, to a cloud infrastructure for analysis of the network-related data. The Key Performance Indicators (KPIs) evaluated are the Received Signal Strength Indicator (RSSI), Signal to Interference plus Noise Ratio (SINR), and Packet Delivery Ratio (PDR). The analysis of these KPIs will help to understand the coverage situation from the IoT device's perspective for the location considered.

1.3 Overview of the Thesis

This thesis is structured to enable the reader to follow the work presented. The following outlines the structure of this thesis:

In chapter 2, the state-of-the-art is discussed with an overview of the LTE Cat-M1 access technology and highlighting the various performance evaluation methodologies. Subsequently, previous works related to LTE Cat-M1 coverage analysis are reviewed presenting the main points, and the approach used in this thesis is mentioned.

In chapter 3, the methodology used in this thesis work is discussed, highlighting the general system architecture and the various components of the system. Subsequently, the software tools used, and the system's end-to-end operational process are explained. Finally, the deployment for measurement campaign is discussed in this chapter.

In chapter 4, the results of the measurement campaign are presented and explained for each device and elevation level.

In chapter 5, discusses the results presented by analysing the results and highlighting the insights obtained. Related future works for possible improvement is also discussed in this chapter.

Chapter 6 summarizes the thesis work.

2 State of the Art

This chapter discusses the general objectives of the LTE Cat-M1 technology and examines its radio resource frame structure and channels for the uplink and downlink radio access. It also reviews the methodology used for performance evaluation as specified by the 3GPP body and the previous research works on coverage analysis for LTE Cat-M1 technology. These related works on coverage analysis generally take either of two distinct approaches: Analysis approached from the evolved Node B(eNB) perspective or analysis from the User Equipment (UE) perspective.

2.1 LTE Cat-M1 Technology Overview

The LTE Cat-M1 is deployed in-band LTE and uses the LTE network as its underlying network. As a result, it inherits some of the features of the LTE network. For example, it uses the same radio uplink and downlink access schemes, frame structure, slot duration, and even channel numerology such as Physical Broadcast Channel (PBCH) [10]. LTE Cat-M1 operates on 1.4 MHz bandwidth, which is the smallest bandwidth of the LTE, but it is distinct from the LTE network and designed to achieve a different set of objectives.

Table 3: Performance objectives for LTE Cat-M1 technology [10] [12]

LTE Objective	Cat-M1	Description
Improved coverage	indoor	The LTE Cat-M1 deployment should achieve an extended coverage of 20 dB relative to the underlying LTE, targeting an MCL of 155.7 dB with a 20 dBm UE (power class 5) and a 164 dB with a 23 dBm UE (power class 3)
Massive number of devices support		Support a relatively large number of devices that are transmitting data with low throughput. 52547 devices per

	cell site sector, implying 157,641 devices per cell site for a cell site covered with three sectors, is estimated by 3GPP
Reduced complexity	LTE Cat-M1 UEs are required to be low-cost devices achieved by reducing the design complexity
Improved power efficiency	Power consumption of LTE Cat-M1 UEs should be reduced to enable support for up to 10 years of battery life with a battery capacity of 5Wh (Watt-hours)
Latency	A 10 second delay can be allowed for applications with strict delay requirement.

Table 3 shows the desired performance objectives for LTE Cat-M1. These objectives are to be achieved without any detriment to the underlying LTE network [10]. The LTE Cat-M1 radio resource must be designed properly to avoid any potential negative impact on the LTE network, alongside other requirements.

2.2 LTE Cat-M1 Radio Resource

1.4, 3, 5, 10, 15, and 20 MHz are the channel bandwidth specified by 3GPP for LTE systems [1]. LTE Cat-M1 operates in the 1.4 MHz LTE channel bandwidth system and utilizes a bandwidth of 1.08MHz for data transmission, corresponding to a maximum of 6 Physical Resource Blocks (PRBs). A PRB is the smallest physical layer radio resource that the eNB allocates to the user device for uplink or downlink transmission, and it comprises a 180 kHz in frequency corresponding to a 0.5ms (1 slot) time duration [13]. The 180 kHz frequency is divided into 12 subcarriers that have 15 kHz spacing between each subcarrier. The PRB bandwidth and subcarrier spacing in the frequency domain is presented in Figure 1.

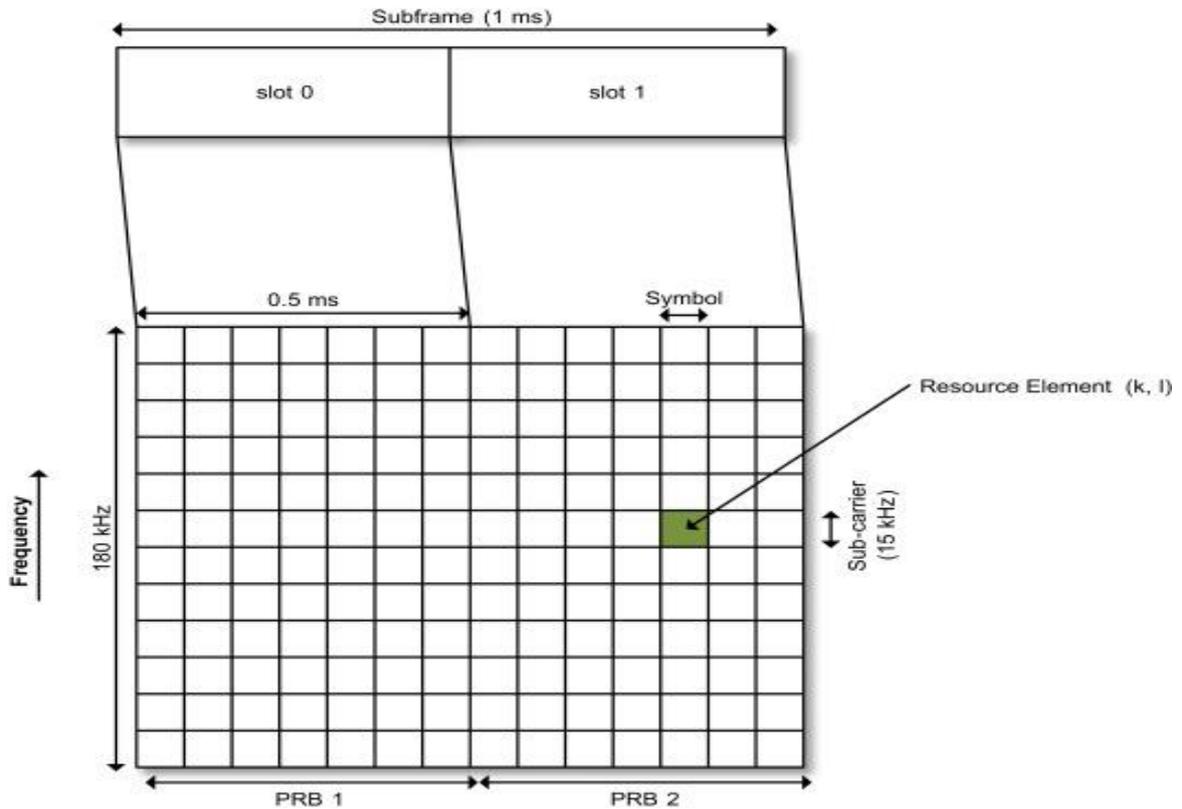


Figure 1: Physical resource block for 1.4 MHz LTE frequency division duplexing [14]

Similar to the LTE, LTE Cat-M1 also uses multiple access technology for uplink and downlink transmission.

2.2.1 Downlink Radio Resource

OFDMA is the radio access technology used in the LTE Cat-M1 system. Using OFDMA allows each of the multiple subcarriers to be modulated with data and inserts a guard band interval between each symbol to prevent inter-symbol interference [15]. Like the time-slot structure in a 1.4MHz LTE network system, LTE Cat-M1 has seven symbols in a time slot of 0.5ms duration and two timeslots in a subframe. As shown in Figure 2, six consecutive subframes make up an M-subframe, which is 6ms in time duration. 10 M-subframes make up an LTE Cat-M1 downlink radio frame, called an M-frame, and have a 60ms time duration [10].

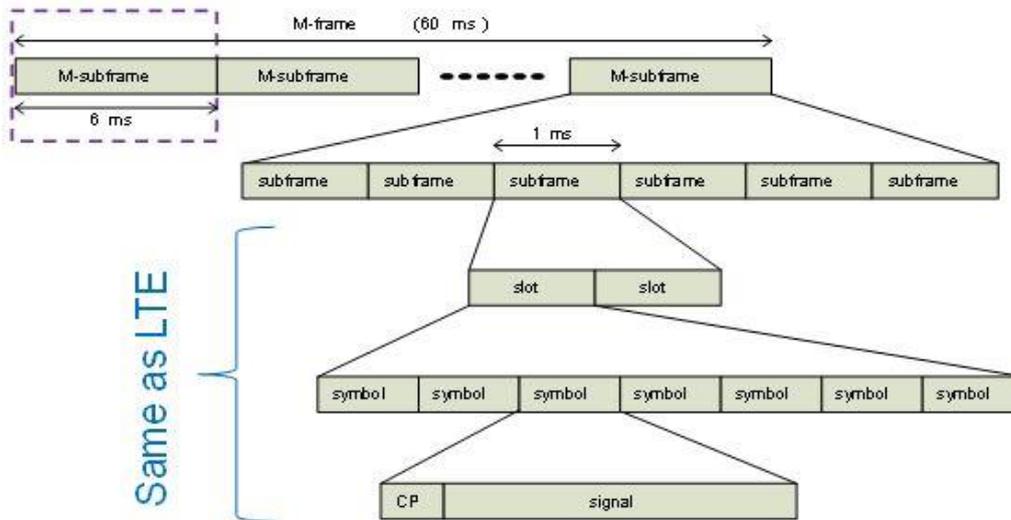


Figure 2: Downlink time structure for LTE Cat-M1 radio access channel [10]

2.2.2 Downlink Physical Channels

The eNB transmits information towards the user device via several downlink physical channels. The channels include LTE-M Physical Broadcast Channel (M-PBCH), LTE-M Physical Downlink Shared Broadcast Channel (M-PDSCH), LTE-M Physical Downlink Control Channel (M-PDCCH), LTE-M Physical Synchronization Channel (M-PSCH). This section extracted from [10] [13] [15].

M-PBCH: This channel is used to broadcast essential cell-specific information for initial cell access. M-PBCH is transmitted on the first 4 symbols in the second slot of each subframe for each of the 6 subframes in the first M-subframe (M-subframe 0) of each M-frame.

M-PDSCH: This channel is used to transmit user device data from eNB to the UE.

M-PDCCH: This channel is used to transmit control information, such as scheduling, ACK/NACK, from the eNB towards the UE. It is transmitted over two OFDM symbols in each of the six subframes in an M-subframe.

M-PSCH: This channel functions to adequately synchronize the time, frequency, and cell ID information, and it comprises the Primary Synchronization Sequence (PSS) and the Secondary Synchronization Sequence (SSS). The PSS helps the UE determine the cell ID within a cell group and obtain the correct subframe timing and frequency offset

information. The SSS enables the UE to determine the cell identity group and the timing of the M-frame.

2.2.3 Uplink Radio Resource

SC-FDMA is the radio access technology used by the LTE Cat-M1 system in uplink transmission. SC-FDMA uses a similar PRB structure as OFDMA but allocates a single subcarrier to a single user for uplink transmission, unlike OFDMA that allows for multiple users to be allocated to a single subcarrier. The subcarrier spacing for the SC-FDMA signal can also be reduced from 15 kHz to 2.5 kHz, in very poor coverage conditions, to accommodate users that presumably do not require high bandwidth [10].

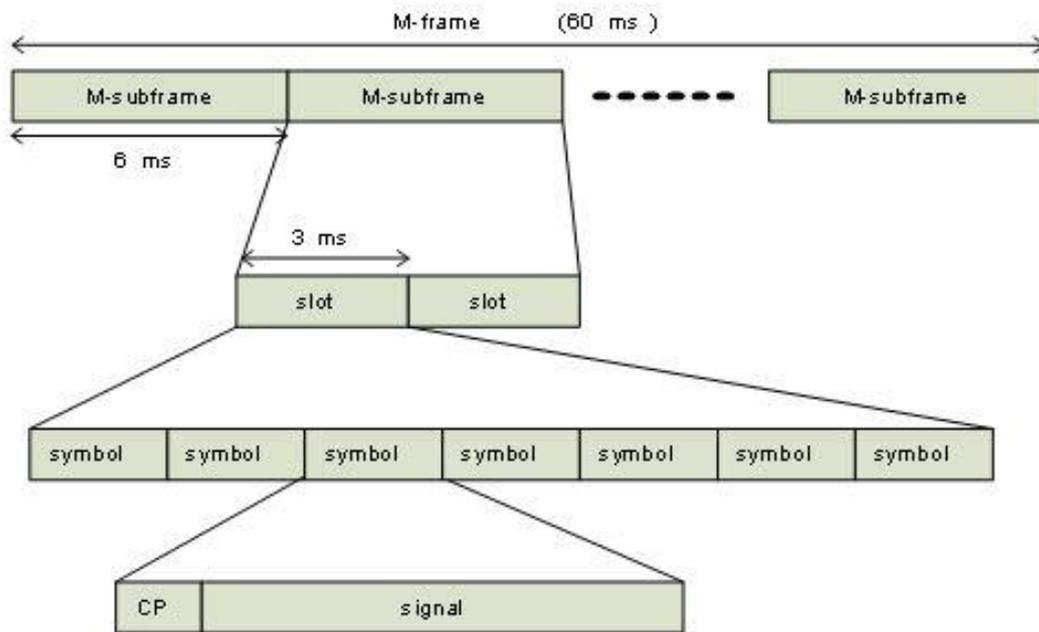


Figure 3: Uplink time structure with 2.5 kHz subcarrier spacing for LTE Cat-M1 radio access channel [10]

As shown in Figure 3, the M-subframe duration is 6ms while the M-frame time duration is 60ms. The timeslot structure for 2.5 kHz subcarrier spacing is different from the 15 kHz counterpart because its M-subframe is composed of two 3ms timeslots, and it has 10 M-subframes in an M-frame [10].

2.2.4 Uplink Physical Channels

The UE uplink transmission to the eNB comprises three main channels: LTE-M Physical Random Access Channel (M-PRACH), LTE-M Physical Uplink Control Channel (M-PUCCH), and LTE-M Physical Uplink Shared Channel (M-PUSCH).

M-PRACH: This channel is used by the UE to request initial access for connection to the eNB and is also used to achieve uplink synchronization for uplink orthogonality [10].

M-PUCCH: This channel is used to transmit uplink control information such as scheduling requests, acknowledgments, and channel quality information [13]. The MPUCCH repetitions can lead to both an increase in power consumption by the UE and a longer delay demodulating the transmitted signal [16].

M-PUSCH: This channel is used to transmits the user payload from the UE to the eNB [17]. The smallest allocation for M-PUSCH transmission, which occurs in very poor coverage conditions, is 1 M-subframe corresponding to 6ms time domain and 2.5 kHz (1 subcarrier) in the frequency domain. Up to 6 UEs are allocated 1 PRB each within an M-subframe for M-PUSCH transmission [10].

2.3 Performance Evaluation Methods

Performance evaluation of the LTE Cat-M1 network is imperative to determine if some of the objectives of LTE Cat-M1 technology are being met. 3GPP specifies some important performance evaluation domains in [10] and [16] that apply to LTE Cat-M1. This section is extracted from [10] [16].

Power Consumption Analysis: This evaluation domain targets the improved power efficiency objective of LTE Cat-M1 technology that seeks to achieve support for up to 10 years of battery life with a battery capacity of 5Wh. There are different power consumption estimates evaluated that use distinct metrics for the various modules within the system. For example, the RF module can be estimated by reception time, UE transmits power and transmission time, and DC power consumed by the power amplifier [16].

Coverage Analysis: This evaluation domain is the focus of this thesis work and, it targets the improved indoor coverage objective of LTE Cat-M1 technology that seeks to achieve an extended coverage of 20 dB relative to the underlying LTE. Coverage performance analysis can be measured from the eNB or UE perspective. MCL value is the key performance indicator (KPI) considered when measuring coverage performance from the eNB perspective [10]. MCL is calculated for both uplink and downlink channels considering repetitions and frequency hopping [10]. The measured MCL is compared with the MCL of the underlying LTE network [16]. However, coverages analysis from

the UE perspective is demonstrated in [18], and this thesis work imitates the same approach as in [18]. The main KPIs considered when analysing coverage performance from the UE perspective are the Signal-to-Interference-plus-Noise Ratio (SINR) and the Received Signal Strength Indicator (RSSI) [18]. Another interesting KPI considered is the Packet Delivery Ratio (PDR).

Capacity Evaluation: The cell spectral efficiency measurement can be used to evaluate the capacity of the system [10]. It can be calculated either through simulation of the system or analytically by comparing to a reference spectral efficiency. The system simulation approach is based on simulations with Mobile Autonomous Reporting (MAR) and Network Command traffic models or based on the software update/reconfiguration model as specified in [10].

2.4 Related Work

The coverage analysis in [19] is done from the eNB perspective and is based on simulation of an actual commercially deployed network configuration in a rural area, with the simulation calibrated using measurements from drive tests and coverage location mapped as in commercial deployment. The results indicate that LTE Cat-M1 even though 99.9% of light indoor users were able to be served, about 20% of deep indoor users did not receive good service. This paper [20] also approaches coverage analysis from an eNB perspective and is based on simulation of a real network deployed in rural and urban areas; but it considers a peculiar situation of LTE Cat-M1 and NB-IoT systems deployed together in dual-network mode. From the results, LTE Cat-M1 shows an improvement by 5dB by supporting links with 161dB MCL compared to the reference 155.7dB MCL specified by 3GPP, but also shows poor performance for deep indoor users with above 165dB MCL. The paper [21] also approaches the coverage analysis from the eNB perspective and is based on simulation with parameters specified in [10]. The results support those in [20] that shows a 5 dB MCL improvement up to 161 dB from 3GPP reference of 156 dB MCL. It also shows that LTE Cat-M1 supports a 164 dB MCL for UEs operating in power class 3 (23 dBm). The results presented in this paper [22], following the eNB perspective, indicates that the uplink transmission of the UE may have a lesser performance than downlink in poor coverage conditions because of the relatively lower transmit power of the UE. It recommends that techniques such as Power Spectral

Density (PSD) boosting may be considered, in addition to repetition technique, to achieve better performance for UEs that need to transmit at a higher data rate in poor coverage conditions.

Many of the previous works reviewed approached LTE Cat-M1 coverage analysis from the eNB perspective, however, this paper [23] takes a hybrid approach to coverage analysis from the UE perspective by conducting field measurement tests to obtain outdoor coverage measurement data and using the data to simulate the signal condition for different indoor coverage levels. The results show a good performance of about 98% coverage for deep indoor users on the LTE Cat-M1 network. The papers [18] and [24] take a pure UE perspective approach to coverage analysis by conducting field investigation of the coverage performance of an already deployed network and measure the SNR and RSSI parameters as the main KPIs. The results show good indoor coverage at different times of day, but for NB-IoT technology.

This thesis work imitates the approach of this paper [18] and the field measurements are conducted in a similar location for some comparison basis. The results in [18] cannot be used as a direct benchmark because it is conducted for different technology. It, however, gives an idea that the results of the measurements in this thesis would likely show a strong indoor coverage performance for LTE Cat-M1 as well, because both technologies are fundamentally very similar.

3 Methodology

This chapter discusses the overview of the system architecture and the relevant protocols used in this system setup. It specifically examines the user-end architecture and the cloud server architecture of the end-to-end system. It also reviews the various protocols used for communication and the individual components of this system while examining the tools used for its development. The end-to-end system design and deployment for field measurement campaign are discussed in the final section of this chapter, highlighting the system operators flow for both User Equipment (UE) and cloud server nodes.

3.1 System Architecture

Figure 4 shows the overall LTE Cat-M1 end-to-end system architecture. This system comprises three essential functional nodes: UE node, telecommunication infrastructure node, and cloud server node. The end-to-end communication between the UE and the cloud nodes was designed using Message Queuing Telemetry Transport (MQTT) IoT protocol.

UE: This is the IoT device installed at several locations within the target area and collects all the relevant user-side data.

Telecommunication Infrastructure: This is the eNB that provides a communication channel through which the IoT devices send data to the cloud. The telecommunication infrastructure used for this work is Telia's LTE Cat-M1 network system.

Cloud Server: The cloud system collects all the data sent from the IoT devices and stores them for further processing or use.

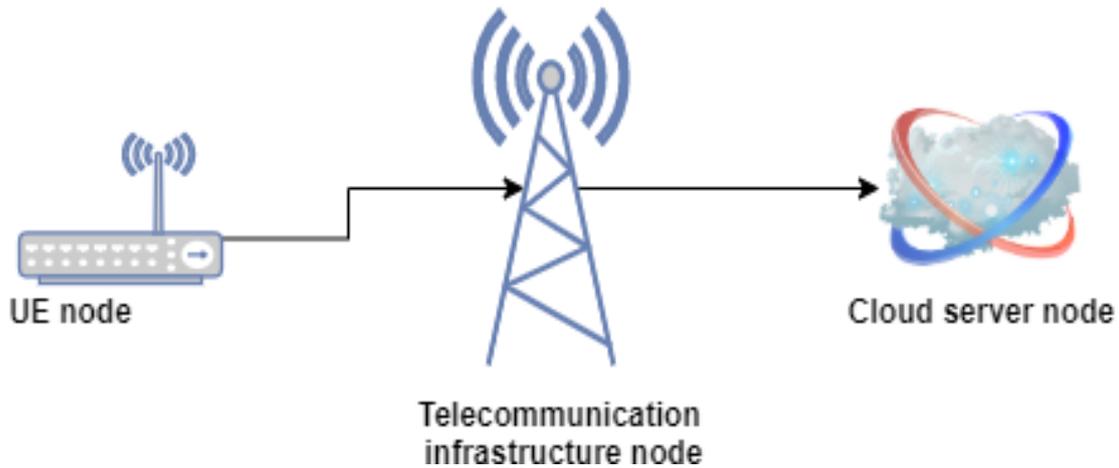


Figure 4: LTE Cat-M1 end-to-end system architecture

Generally, the UE node first establishes an initial network connection to the LTE network and then attaches to the Cat-M1 network services. It subsequently establishes a secured connection with the cloud server. After setting up an end-to-end connection from the UE node to the cloud server node, the IoT devices periodically collect all relevant network data and send them, together with the image snapshot, to the cloud server, which stores the network data for further processing. The functional components of the UE and cloud server nodes are explained in the following sections.

3.1.1 UE Node

This node comprises the camera and the communication modules, which are OpenMV Cam H7 and Avnet Silica NB-IoT Sensor Shield devices, respectively.

Camera Module: The OpenMV Cam H7 device is a machine vision camera that is programmable, low-powered, and has a small form factor. It has an embedded microcontroller board, which includes several features, and uses MicroPython [25] operating system that makes it possible for the board to be programmed in MicroPython programming language [26]. The microcontroller (MCU) board on the OpenMV Cam H7 is an STM32H743VI device [26] built on a high-performance Arm Cortex-M7 32-bit Reduced Instruction Set Computer (RISC) core that operates on a 480 MHz frequency and SRAM and flash memory of 1MB and 2MB respectively [27]. It has 10 Input/Output (I/O) pins that have an output voltage of 3.3V but are also tolerant to 5V. As shown in Figure 5, the OpenMV Cam H7 camera module also has a USB interface through which it can also be programmed, a microSD card socket for up to 64GB external microSD card storage; an SPI bus that allows for image data streaming to an external component; and

an I2C, CAN and UART buses for interfacing with external components and devices. It also has a 12-bit ADC and DAC, a total of three I/O pins used for controlling servo motors, interrupts and PWM on all 10 I/O pins, and RGB and IR LEDs [26].

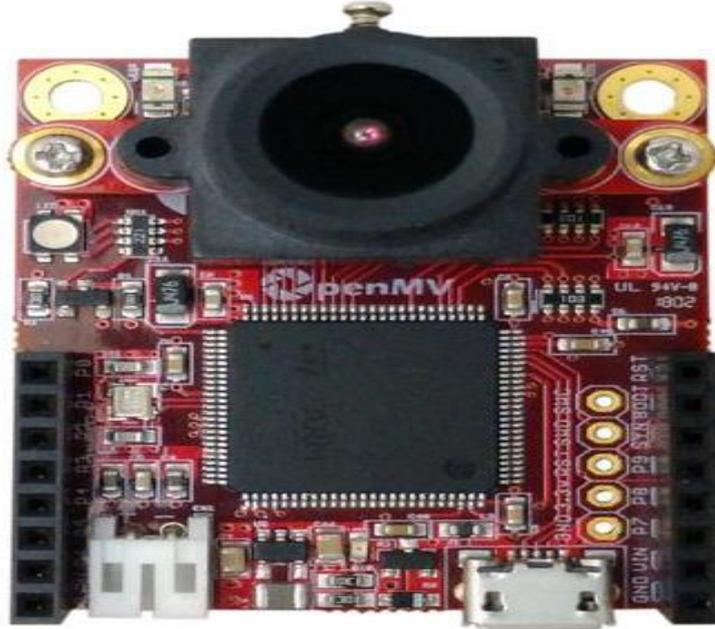


Figure 5: OpenMV Cam H7 device [26]

The image sensor on the OpenMV Cam H7 is a high-performance, low-power OV7725 that can function in temperatures from -20°C to $+70^{\circ}\text{C}$. This user-controllable image sensor has a 2.8mm adjustable lens and supports a 640x480 image array that can operate at 60 frames per second in VGA mode. This camera module can be used in home automation, robot guidance, industrial applications, surveillance application, object detection and tracking applications [26].

Table 4: OpenMV Cam H7 pinouts description [28]

Pin			Description
Header	No	Name	
J1 Pin Configuration			
J1	1	P0	UART1 RX – TM1 CH3N – SPI 2 MOSI
	2	P1	UART1 TX – TM1 CH2N – SPI 2 MISO

	3	P2	CAN2 TX – TM1 CH1N – SPI 2 SCLK
	4	P3	CAN2 RX – SPI 2 SS
	5	P4	TIM2 CH3 – I2C 2 SCL – UART 3 TX
	6	P5	TIM2 CH4 – I2C 2 SDA – UART3 RX
	7	P6	TIM2 CH1 – DAC – ADC
	8	3.3	3.3V Rail (250 mA Supply MAX).
J2 Pin Configuration			
J2	1	RST	Reset (Connect to GND to reset)
	2	BOOT	Boot 0 (Connect to 3.3V for DFU mode)
	3	SYN	Frame synchronization pin (Use to frame sync cams)
	4	P9	Servo3 – TIM4 CH3
	5	P8	Servo2 – TIM4 CH2 – I2C4 SDA
	6	P7	Servo1 – TIM4 CH1 – I2C4 SCL
	7	VIN	VIN (3.6V – 5V)
	8	GND	GND Rail
J3 Pin Configuration			
J3	1	SWC	Serial wire debug clock
	2	SWD	Serial wire debug data
	3	RST	Reset (active low)
	4	3.3V	3.3V rail (500 mA Supply MAX)

	5	GND	GND rail
--	---	-----	----------

Table 4 shows the I/O pinouts of the OpenMV Cam H7 camera module and their description. For this thesis work, P4 and P5 pins are used for UART serial communication with the communication module, while 3.3 and GND pins are used for power supply connections.

Communication Module: The Avnet Silica NB-IoT Sensor Shield is based on the Quectel BG96 module that supports NB-IoT and Cat-M1 cellular IoT communication technologies. This modem includes a SIM card slot for connection to the telecommunication network (Telia's LTE Cat-M1 network) and can be controlled simply by issuing Attention (AT) commands [29], which are a set of instructions for communication modem's control [30]. The Avnet Silica NB-IoT Sensor Shield is compatible with the Arduino UNO board and has Peripheral Module (Pmod) connectors that present the possibility to interface with several other boards, improving its flexibility. There are several types of Pmod interfaces; this shield uses the Pmod interface type 4A (expanded UART) that incorporates an optional hardware flow control signal [31] [32]. The Pmod connectors on this shield are shown in Figure 6, and the pinouts description are presented in Table 5.

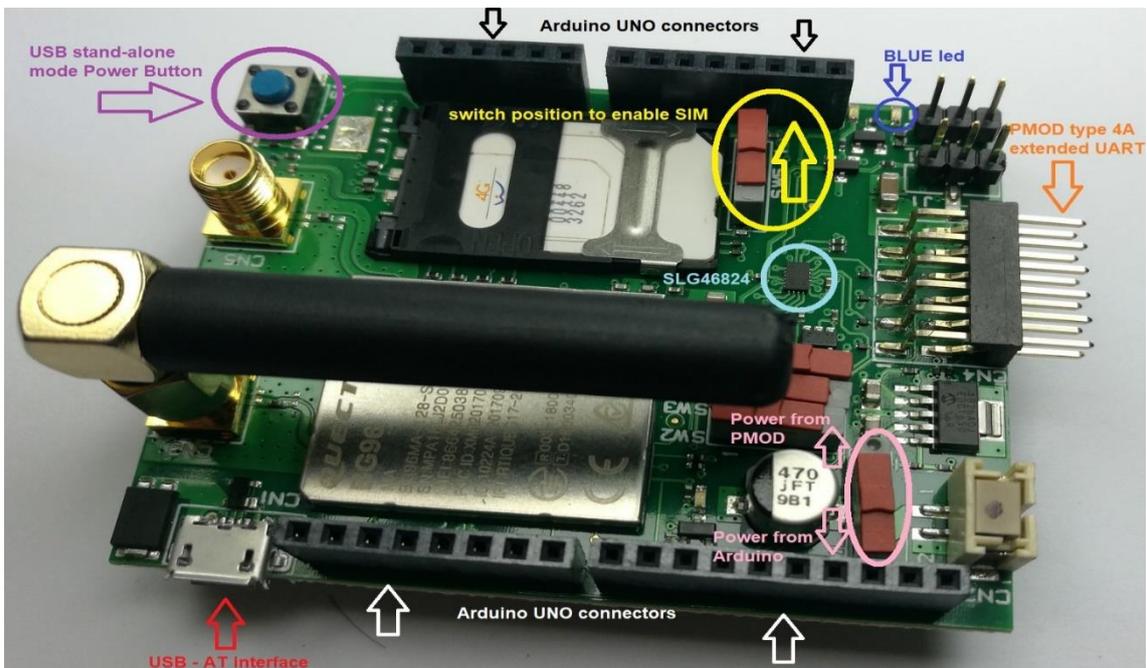


Figure 6: Avnet Silica NB-IoT Sensor Shield [31]

Table 5: Avnet Silica NB-IoT Sensor Shield pinouts description [31]

Pin	Signal	Description
1	UART0_CTS_3V3	Clear to send
2	UART0_RXD_3V3	UART receive
3	UART0_TXD_3V3	UART transmit
4	UART0_RTS_3V3	Request to send
5	GND	Ground
6	VCC_3V3_PMOD	3.3V supply
7	BG96_STATUS	Interrupt signal
8	RESET_PMOD	Reset signal
9	NC	Not connected
10	NC	Not connected
11	GND	Ground
12	VCC_3V3_PMOD	3.3V supply

In this thesis work, the Pmod connectors were used to physically interface with the OpenMV Cam H7 as illustrated in Figure 7.

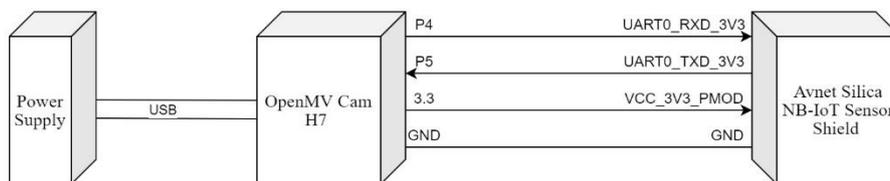


Figure 7: Schematic diagram of the UE node connections

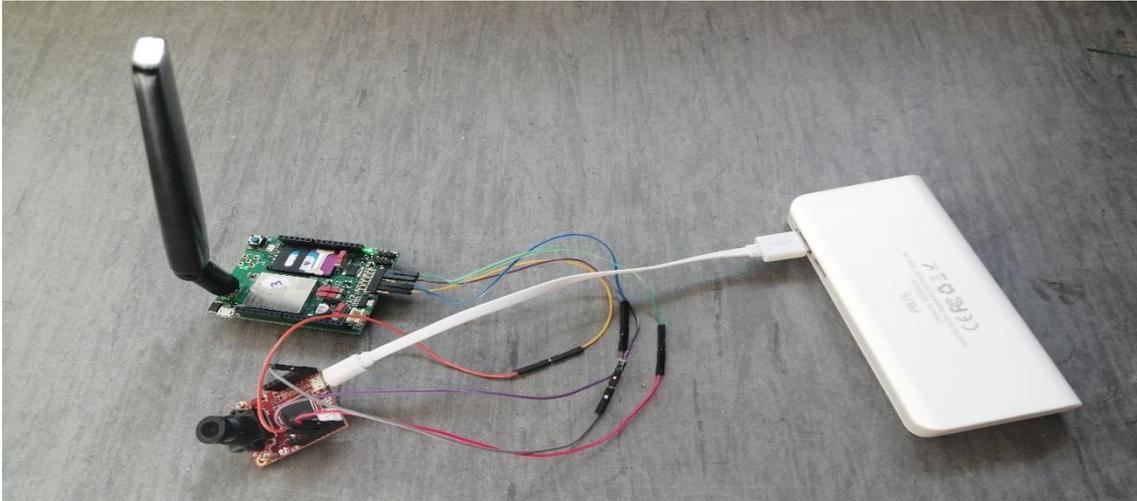


Figure 8: Complete UE node

Figure 8 shows the complete UE node with its intricate components connected. The camera module is powered with an external power supply unit connected to the USB interface of the module and consequently powers the communication module via its 3.3 and GND pins connected to the VCC and GND pins of the communication module, respectively. The UART connection between both modules was used to establish a serial communication in both directions for data transfer. The UE node also serves as the MQTT publisher, which is an MQTT client that sends IoT data to the MQTT broker.

3.1.2 Cloud Node

The cloud node comprises an MQTT broker and MQTT subscriber modules; all deployed on individual cloud virtual machines. The cloud virtual machines was Amazon Web Services (AWS) Elastic Compute Cloud (EC2) service running on Ubuntu 20.04 distribution of the Linux operating system.

MQTT Broker: The MQTT broker, also called the MQTT server, is a component of the MQTT protocol discussed in further detail in section 3.1.3. The MQTT broker simply serves to receive messages from MQTT clients published on one or more MQTT topics. It then filters these messages and forwards them to other MQTT clients that have subscribed to the specific topic [33]. Only clients that subscribe to the topic can receive messages published on that topic.

There are three deployment options for the MQTT broker: local installation, managed or online/shared application [34]. The local installation deployment option was used for this thesis because it provides more flexibility, especially regarding detailed packet capture

for deeper performance analysis. There are several MQTT broker software tools, but some popular ones are Mosquitto and HiveMQ [35]. Mosquitto broker was used in this system because it is open source, has rich documentation, and good online support. The Mosquitto MQTT broker version 3.1.1 was installed on an Ubuntu 20.04 Linux virtual machine using the following standard commands:

- `sudo apt-get update`
- `sudo apt-get install mosquitto`
- `sudo apt-get install mosquitto-clients`

All network data traversing the network interface of the MQTT broker was captured and saved on the machine using `tcpdump` tool. The `tcpdump` command used to collect and save the data with changes in the name file is as follows:

- `sudo tcpdump -i ens5 -w u01_27-04-2021_morn.pcap`

The function of the MQTT broker in this system architecture is to receive data from the UE node and route it to the MQTT subscriber.

MQTT Subscriber: The MQTT subscriber is discussed further in section 3.1.3. Essentially, it is an MQTT client that receives messages sent to the broker on the specific topic to which it has subscribed [35]. Subscription is done to the topic on the MQTT broker, and the MQTT subscriber can only receive data on those topics. An MQTT subscriber can be any device or application that supports MQTT protocol, such as a server application. The subscriber used in this thesis was an Ubuntu 20.04 Linux virtual machine running a Python script running written to subscribe to the topic of interest to collect and save the sent IoT network data. The Python script is provided in Appendix 2.

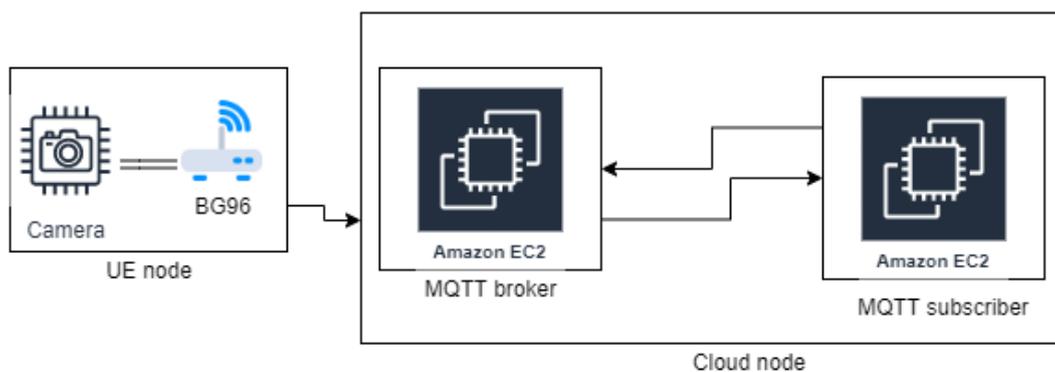


Figure 9: Schematic of the end-to-end system architecture

The schematic of the cloud node is shown in Figure 9. The UE node connects to the MQTT broker and points to its IP address as the next hop for data transfer. This node collects and transfers the image and signal strength data to the broker on separate topics. It also captures the network traffic during communication with the UE node. Subsequently, the broker forwards the data to the MQTT subscriber already subscribed to the topics.

3.1.3 Communication Protocols

The components used in this system support a wide variety of protocols for communication and transferring IoT data between the device and the cloud. However, the protocols used for implementing this system design are Transmission Control Protocol (TCP), Message Queuing Telemetry Transport (MQTT), Universal Asynchronous Receive-Transmit (UART).

TCP: The TCP is a full-duplex connection-oriented transport protocol designed to enable end-to-end and reliable communication between connected hosts in a network [36]. In the TCP/IP stack, as shown in Figure 10, TCP protocol enables data transmission between the Application layer and the Internet layer through network sockets connection that it first establishes and maintains throughout the data transmission [37]. This commonly used protocol incorporates the Positive Acknowledgement and Retransmission (PAR) protocol to ensure end-to-end and accurate data delivery but sometimes lead to delivery delays that make TCP less suitable for real-time applications where speed of delivery is crucial [37] [38].

Applications built on the TCP/IP stack and using TCP as the transport layer protocol send data streams to the TCP layer. The protocol packages these data streams into small blocks and formats them into TCP segments by adding TCP headers [37]. TCP segments are the basic data units of the TCP protocol and comprise a header and a data section as shown in Figure 11.

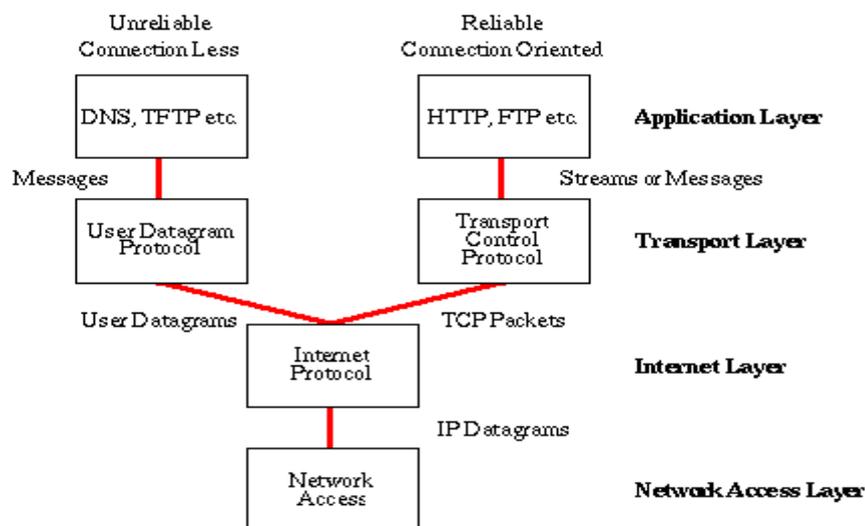


Figure 10: TCP/IP protocol stack [37]

TCP Header				
Bits	0-15			16-31
0	Source port			Destination port
32	Sequence number			
64	Acknowledgment number			
96	Offset	Reserved	Flags	Window size
128	Checksum			Urgent pointer
160	Options			

Figure 11: TCP segment header structure [39]

The host-to-host communication of TCP involves three steps: establishing a connection, transmitting data in the TCP segment, and closing the connection. As illustrated in Figure 12, TCP establishes an end-to-end connection using a three-way handshake process where the requesting client first sends an SYN (Synchronize) segment to the server with a random segment sequence number. The server agrees to this connection request by replying with an SYN-ACK (Synchronize-Acknowledge) segment, increments the client's random segment sequence number by 1, and generates its own random segment sequence number. Subsequently, the client then replies to the server with an ACK segment and increments both sequence numbers by 1 [39].

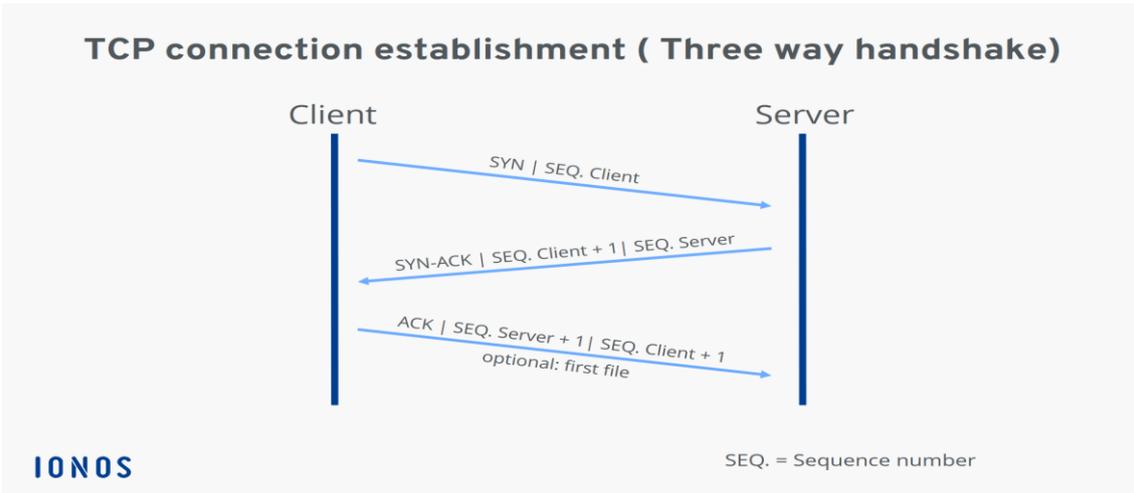


Figure 12: TCP three-way handshake for connection establishment [39]

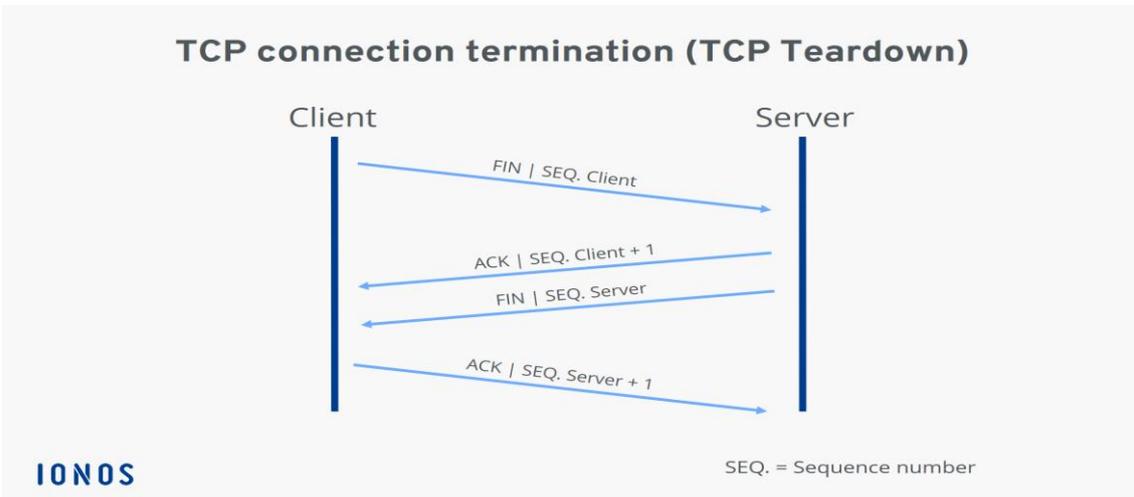


Figure 13: TCP connection termination [39]

After establishing the connection, the data can be transferred between the two hosts with a sequence number for each transmitted data. The recipient sends an ACK after receiving each data. Finally, as illustrated in Figure 13, either the client or server initiates the connection closure by sending a FIN (Finish) segment to the other end and its sequence number. The other end replies with an ACK segment with sequence number incremented by 1 and a FIN segment; then the initiating end sends an ACK segment with an increment in sequence number by 1 [39].

MQTT: MQTT is a simple bi-directional messaging protocol that is lightweight, scalable, and flexible; and operates on a publish and subscribe model [40] [35]. This protocol, first developed in 1999, is increasingly being preferred for IoT implementations [41]. It is event-driven, which implies that the protocol is called to action when the relevant event has occurred [35]. This protocol is based on the publish-and-subscribe

model, where the publish operation (performed by the publisher) involves sending an MQTT message (IoT sensor data) to a topic; and the subscribe operation (performed by the subscriber) involves receiving the MQTT message published to a topic. MQTT topics, themselves, are a hierarchical way of specifying an address for the MQTT message payload to be transmitted; and uses UTF-8 strings, are case sensitive, and must consist of at least one character [33]. IoT systems based on MQTT comprise two essential components: MQTT broker, or server, and MQTT client. The MQTT broker filters and forwards the messages it receives on a topic to all subscribers to that topic whereas, the MQTT client can publish and/or subscribe to a topic [33]. As shown in Figure 14, the MQTT client publishes IoT sensor data as an MQTT message on a topic to the MQTT broker, which in turn distributes the messages to other clients that have subscribed to the topic. The MQTT broker typically does not store messages but supports three Quality of Service (QoS) levels, QoS 0, 1, 2, that have message buffering capability for reliable delivery [35].

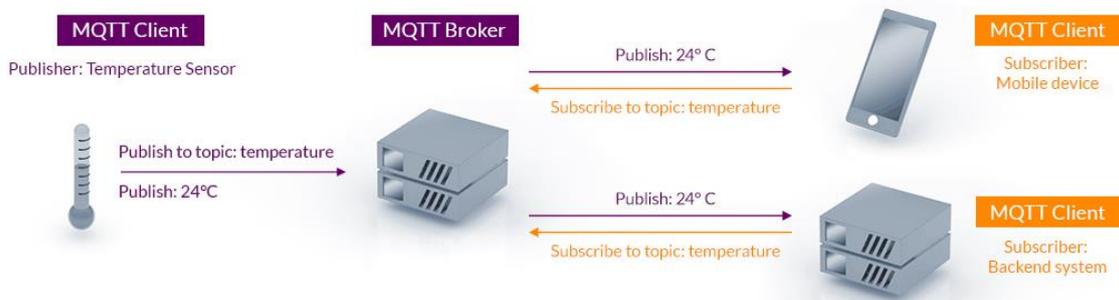


Figure 14: MQTT publish/subscribe architecture [42]

Table 6: Types of MQTT control packet [43]

Name	Value	Direction of flow	Description
Reserved	0	Forbidden	Forbidden
CONNECT	1	Client to Server	Client request to connect to Server
CONNACK	2	Server to Client	Connect acknowledgement
PUBLISH	3	Client to Server / Server to Client	Publish message

PUBACK	4	Client to Server / Server to Client	Publish acknowledgement
PUBREC	5	Client to Server / Server to Client	Publish received (assured delivery part 1)
PUBREL	6	Client to Server / Server to Client	Publish release (assured delivery part 2)
PUBCOMP	7	Client to Server / Server to Client	Publish complete (assured delivery part 3)
SUBSCRIBE	8	Client to Server	Client subscribe request
SUBACK	9	Server to Client	Subscribe acknowledgement
UNSUBSCRIBE	10	Client to Server	Unsubscribe request
UNSUBACK	11	Server to Client	Unsubscribe acknowledgement
PINGREQ	12	Client to Server	PING request
PINGRESP	13	Server to Client	PING response
DISCONNECT	14	Client to Server	Client is disconnecting
Reserved	15	Forbidden	Reserved

MQTT protocol is a binary-based protocol that uses the TCP/IP protocol stack with TCP as its transport protocol for client-broker connectivity [33]. Leveraging TCP transport MQTT uses a command and command acknowledgment format, which ensures that an acknowledgment is sent by the broker, for every command sent by the client [43]. As shown in Table 6, the MQTT protocol has 16 control packet types, although a few are used in most implementations [33].

UART: UART is a simple hardware communication protocol with a configurable speed used to transmit and receive serial data asynchronously [44]. This protocol uses less circuitry and wires, which helps to reduce implementation costs. For example, as shown

in Figure 15, UART requires only two wires to be connected, on dedicated transmitter and receiver pins, between the hardware devices to achieve full-duplex communication [44]. In the UART connection illustrated in Figure 15, the transmitter pin of the transmitting UART connects to the receiver pin of the receiving UART, and the transmitter pin of the receiving UART connects to the receiver pin of the transmitting UART, indicating how data flows between the transmitting and receiving UARTs [44] [45]. The transmitting and receiving UARTs also need to be configured with the same baud rates.

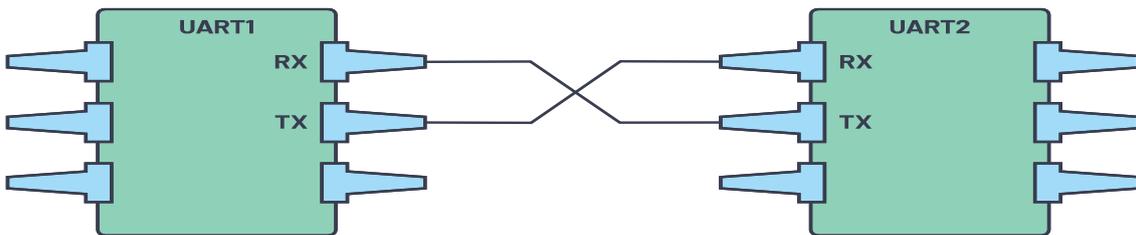


Figure 15: UARTs connection for serial data transmission [44]

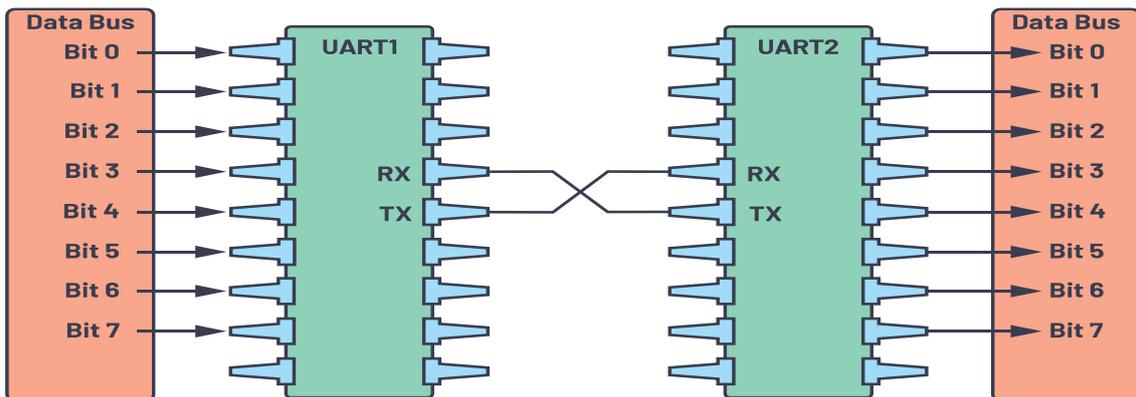


Figure 16: UARTs connection with data bus

In UART communication illustrated in Figure 16, the transmitting UART obtains the data in parallel format on the data bus from a connected controlling device and converts it into a serial form that it transmits bit-by-bit to the receiving UART. The receiving UART does the reverse process by converting the serial data back into its parallel format and sending it to its connected device [45].

3.2 System Development Tools

The configuration of the end-to-end system was achieved using several tools. The most relevant ones include OpenMV Integrated Development Environment (IDE), QCOM

software, Terminal emulator software, Wireshark and tcpdump, and Amazon Web Services (AWS) console.

OpenMV IDE: The OpenMV IDE, developed by OpenMV, is the main integrated development environment for programming the OpenMV Cam. It is an easy, powerful text editor with debugging terminal and frame buffer with a histogram display [46]. The IDE incorporates many relevant examples for a variety of use-cases such as face detection, colour tracking, machine learning, etc. The OpenMV IDE is an efficient tool for programming the OpenMV Cam as several actions can be performed with a few clicks. For example, uploading the code to the OpenMV Cam can be achieved by clicking on Tools - Save the open script to your OpenMV Cam. After physically connecting, the device can be accessed for writing and uploading codes by clicking on the connected button on the OpenMV IDE. This IDE has a frame buffer viewer that conveniently displays the camera's capture in real-time and the text editing environment simultaneously, providing the ability to immediately observe the effect of the camera's view as code is written [47].

QCOM: QCOM software tool is provided by Quectel and does not require an installation to run. This tool is used to send AT commands via serial COM port that can be configured with settings such as COM port, baud rate, stop bits, parity, byte size, flow control. This tool can also be used to send files to the Quectel BG96 and can save logs. QCOM software enables the configuration of the connecting device using AT commands [48].

Terminal Emulator: The main terminal emulator tool used in this thesis work was Putty. Putty is an open-source, lightweight terminal emulator software that supports several network protocols such as SSH, Telnet, Serial, SCP, raw socket connection. This tool interface is based on a command line and is used to connect and administer devices and transfer files to remote devices. The Putty software tool was developed by Simon Tatham in 1999 and can work on various operating system platforms for both 32bits and 64bits [49]. Another important terminal emulator tool used for this thesis work was Solar-Putty. Solar-Putty software is a similar software tool just as Putty but was crucial in this thesis because of its support for multiple tabbed sessions and its feature that allow users to save session login credentials for easy re-login [50].

Wireshark and Tcpdump: Wireshark is an open-source network packet analyser Graphical User Interface (GUI) tool used to capture live packet data in detail for network troubleshooting, communication protocol debugging, security analysis, etc. [51]. This tool was initially developed in 1998 by Gerald Combs and has support for various operating system platforms. It can capture several network communications protocols on a live network connection and display the protocols in hierarchies. The Wireshark tool captures all network traffic traversing the network interfaces and can decode the captured packets [51]. It also has several other features like deep protocol inspection, packet filtering, etc. A similar tool also used for this thesis work is the tcpdump tool. Tcpdump also captures packets flowing through a network interface, but it is a non-GUI tool. In this thesis work, tcpdump was install in the MQTT broker component of the cloud node to capture the packets flowing between the UE and cloud nodes and save them in a suitable format. These captured packets were then analysed using Wireshark.

AWS Management Console: The AWS management console is a web application used to manage services and infrastructure deployed on the AWS cloud [52]. The console itself includes separate consoles for managing each AWS cloud service, making it a single interface that embodies and provides access to all other services [53]. In addition to being a web application, this single interface makes it easy to access the AWS console because there is no need for software installation effort, and convenient to use AWS cloud services. This tool can run on various devices and operating system platforms, including mobile operating systems [52].

3.3 System Design and Deployment

The prerequisite to the design of the system used in this thesis was an evaluation of the features and protocols supported by each module in comparison to the overall thesis objective. The protocol used and design approach was based on the common industry practice. It is noteworthy that several end-to-end IoT system designs were tested to achieve the final system; however, the final design was chosen based on the ability of the system to effectively acquire and transfer the image and collect detailed network-related data for analysis.

3.3.1 UE Node Design

The main objective of this node is to acquire image and signal strength information and transfer all this information to the cloud through the telecommunication infrastructure. To achieve this objective, the node must first connect to the telecommunication network; then connect to the cloud node, which is its next hop for the data transfer. Also, Telia's telecommunication infrastructure was provisioned to allow for the connection of the communication module to the LTE Cat-M1 network using specific SIM cards. The UE node was mainly developed using MicroPython codes for the camera module and some of these codes also include AT command instructions to the communication module via UART communication protocol. The software program for this node, shown in Appendix 3, was written on OpenMV Integrated Development Environment (IDE), discussed in section 3.2, and uploaded on the OpenMV Cam module.

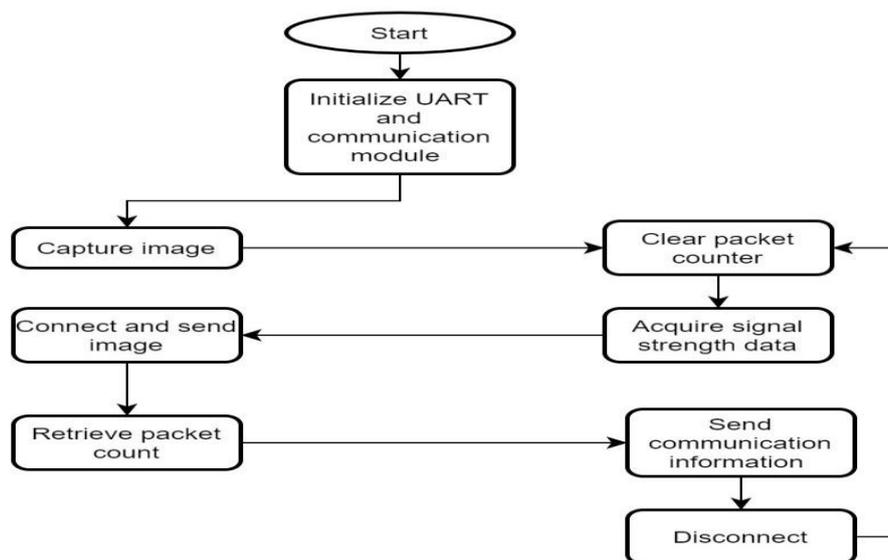


Figure 17: UE node program flow

As shown in the node's software program flow in Figure 17, the program begins by initializing the UART serial communication bus responsible for communication between the two modules in the UE node. The communication module was subsequently initialized to operate as an LTE Cat-M1 UE, connect to the LTE Cat-M1 network, and attach to the appropriate Access Point Name (APN) on Telia's telecommunication network for packet data service. The camera module then captures the image and notes its size for transfer. The program enters a loop where it clears the packet data counter on the communication module and acquires the signal strength information. It then sends the previously acquired image to the cloud node and counts the data network packets. Subsequently, it sends all

the acquired communication information like signal strength, packet count, timestamp, and image size to the cloud node.

3.3.2 Cloud Node Design

The main objective of the cloud node is to receive data from the UE node. The MQTT broker was configured by issuing the commands in section 3.1.2. The steps for the packet mirror configuration on AWS cloud are as follows:

- After setting up the EC2 instance for the packet mirror function, log in to the VPC console for your MQTT broker's EC2 instance.
- Create a mirror target.
- Create a mirror filter.
- Create a mirror session.
- Finally, log in to the EC2 instance and issue the command in section 3.1.2 to start capturing packets.

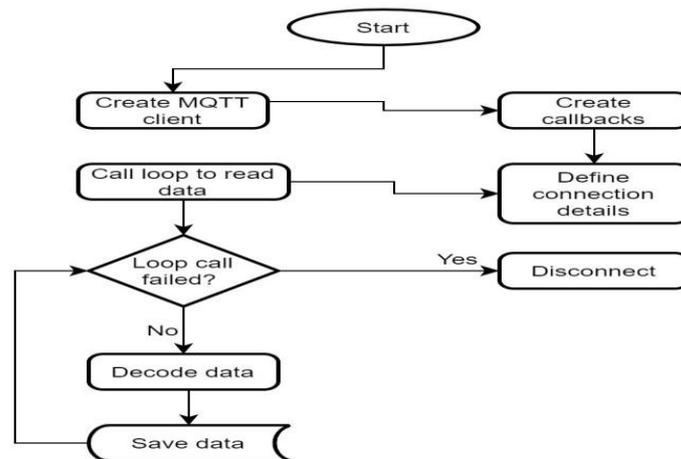


Figure 18: MQTT subscriber program flow

The MQTT subscriber's Python script and program flow are presented in Appendix 2 and Figure 18, respectively. The program begins by creating an MQTT client and subsequently creating callback functions that run automatically when the client connects to the broker and when it receives a message from the broker. Then, the program defines the connection parameter and MQTT topic subscription. It then enters a continuous loop where it periodically reads the data on the MQTT broker's buffer for that topic. This data is finally saved on the MQTT subscriber. The reference codes for the MQTT subscriber are [54] [55].

3.3.3 Deployment

Eight UE node devices were deployed indoors in a building area within TalTech main building for the field measurement campaign. The devices were installed closed to the window on four floors (elevation levels), with two devices per floor. Each floor is approximately three meters apart in height. The UE node devices were installed approximately between 145 to 179 meters away from the base station, which is installed at a height of 20 meters from the ground. The summary of the approximate distance of each device from the base station is presented in Table 8, and the map-view of the devices installed is shown in Figure 19.

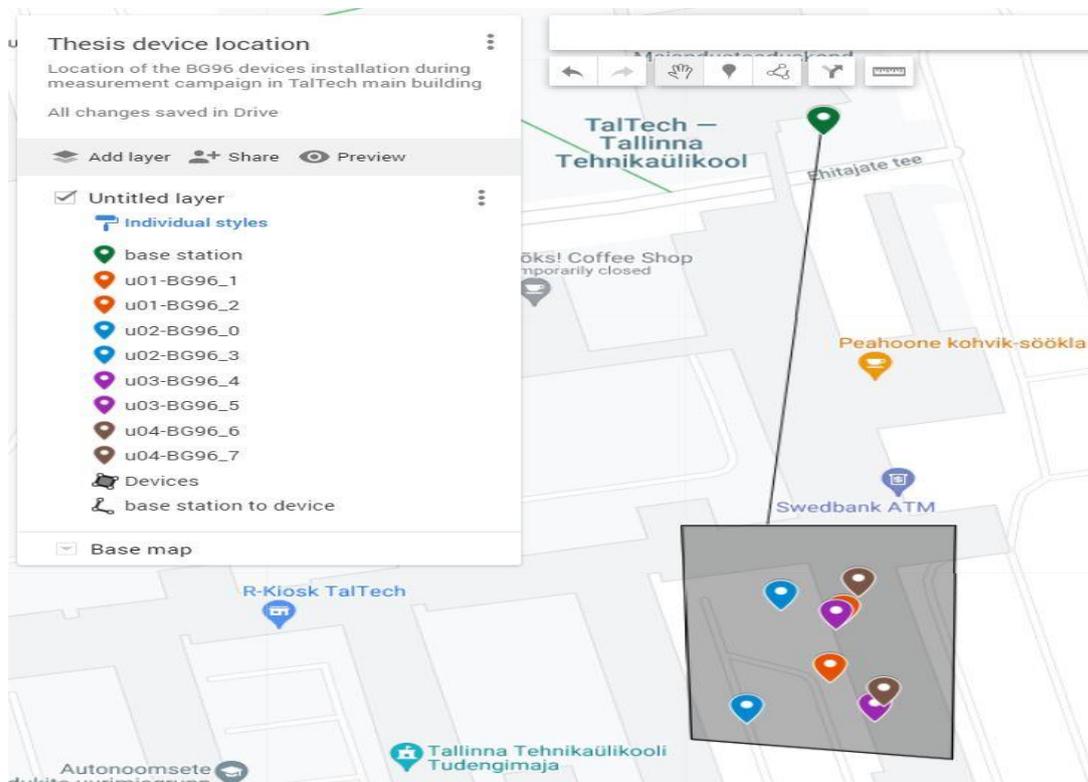


Figure 19: Map view of devices installed.

Table 7: Approximate distance and elevation of each device installation during campaign

Floor Number	Device ID	Distance to Base Station (m)	Elevation from Ground (m)
1	BG96_1	150	0
	BG96_2	167	0

2	BG96_0	146	3
	BG96_3	179	3
3	BG96_4	151	9
	BG96_5	177	9
4	BG96_6	145	12
	BG96_7	177	12

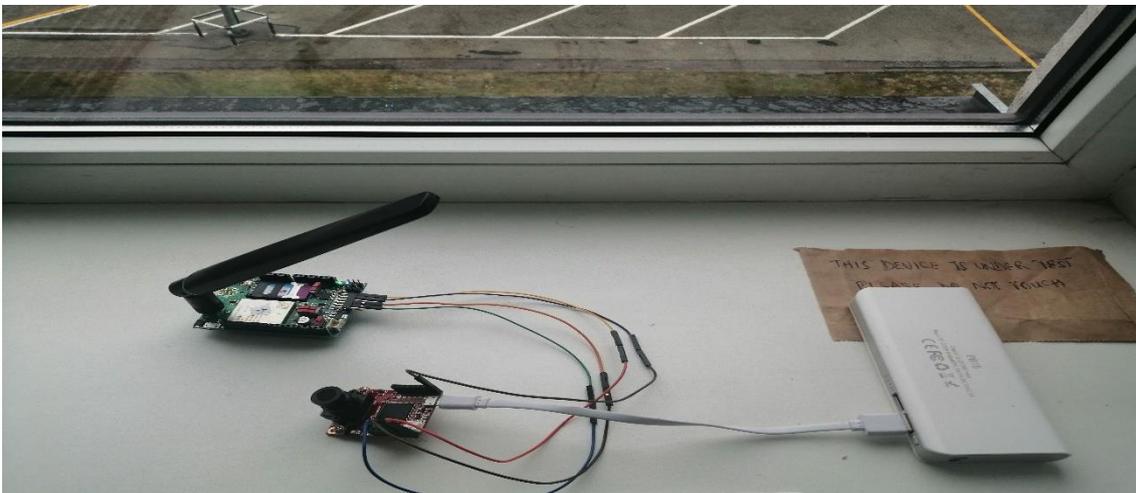


Figure 20: UE node installed on a floor.

Figure 20 shows an example of a UE node device installed on one of the floors during the measurement campaign. Measurements were taken periodically from the installed devices for a total duration of 12 hours (6 hours per period) from 08:00 to 14:00 GMT+3 (morning period) and from 17:00 to 23:00 GMT+3 (evening period) during the campaign.

4 Measurement Results

The results of the measurement campaign conducted for the four floors is presented in this chapter. It includes the signal strength graphs, summary table of the signal strength on each floor, and packet delivery ratio table. The signal graphs presented for each floor comprises the UE node level and floor level graphs.

The results show the plot of the RSSI and SINR signal parameters for the devices installed on each of the floors and the general RSSI and SINR for each floor during morning and evening hours. The dotted lines are the data trendlines that show the pattern and direction of the data. Each signal parameter has a trendline with the corresponding colour and the results generally help understand the coverage performance for each floor location.

Table 8: NB-IoT Signal Strength (RSSI) reference values [18]

LTE	NB-IoT	Signal Strength
> -65 dBm	> -60 dBm	Excellent
-65 to -75 dBm	-60 to -80 dBm	Good
-75 to -85 dBm	-80 to -95 dBm	Fair
-85 to -95 dBm	-95 to -110 dBm	Poor
< -95dBm	< -110 dBm	Disconnect

Table 8, used in this paper [18], shows the reference values indicating the performance level and these values also applies in this thesis work.

4.1 First Floor

The Figures 21 to 24 show the results of the measurement campaign on the first floor. These results represent the signal strength performance for each UE node device on the first floor and the general signal strength for all the devices on the first floor.

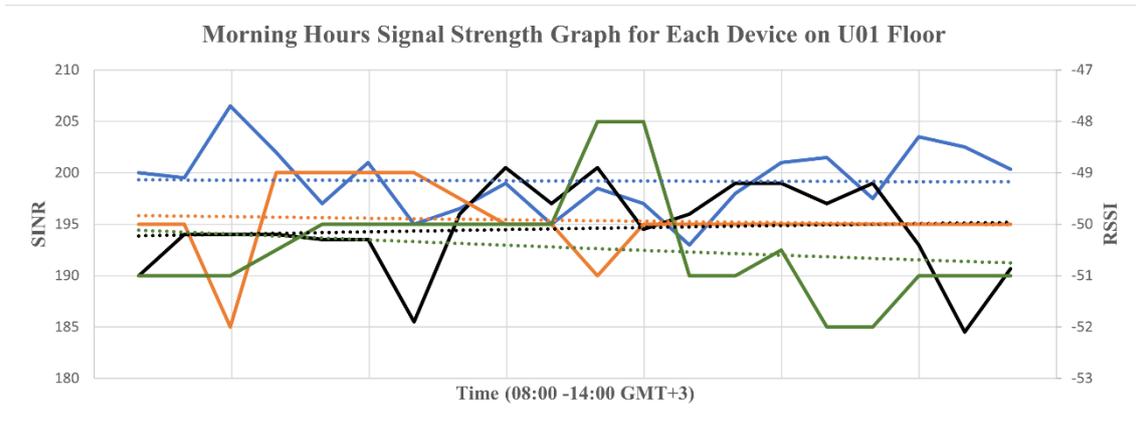


Figure 21: Signal strength graph for each UE node device on the first floor during morning hours

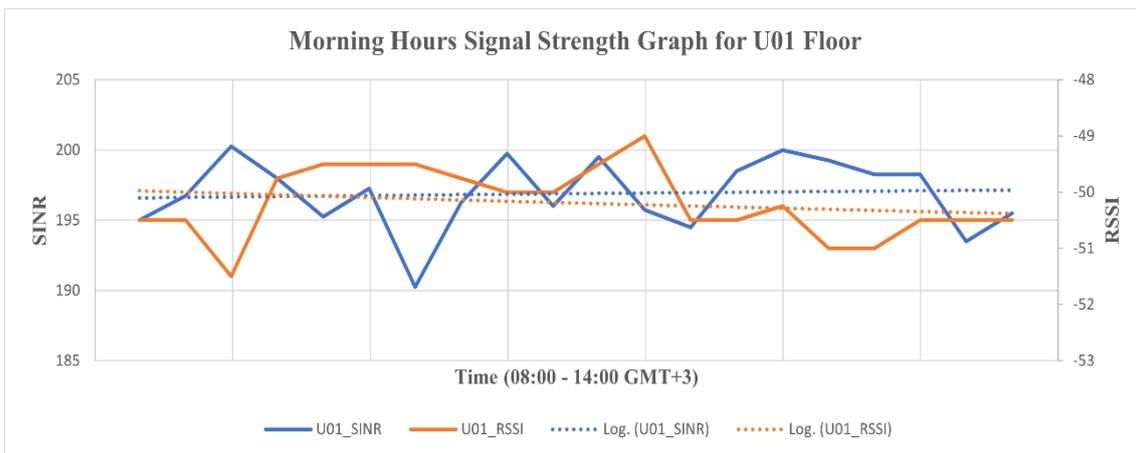


Figure 22: Signal strength graph for the first floor during morning hours

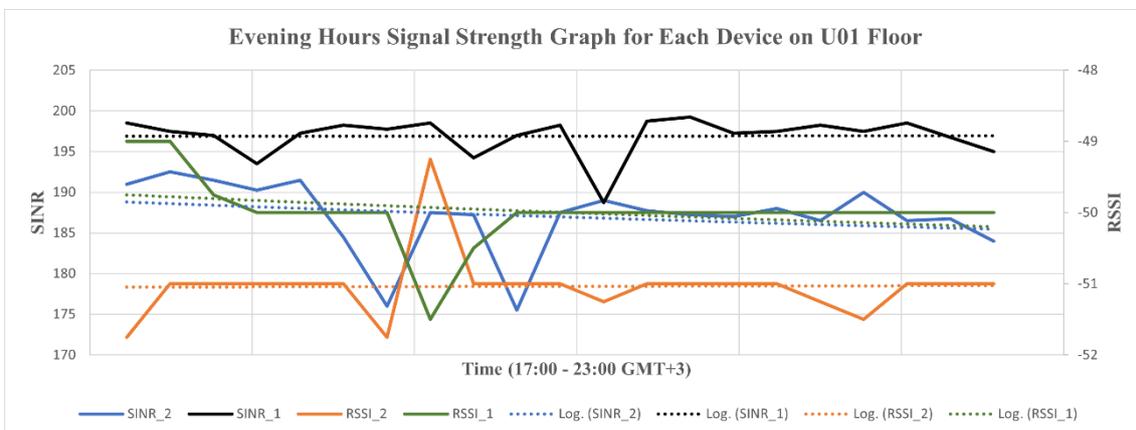


Figure 23: Signal strength graph for each UE node device on the first floor during evening hours

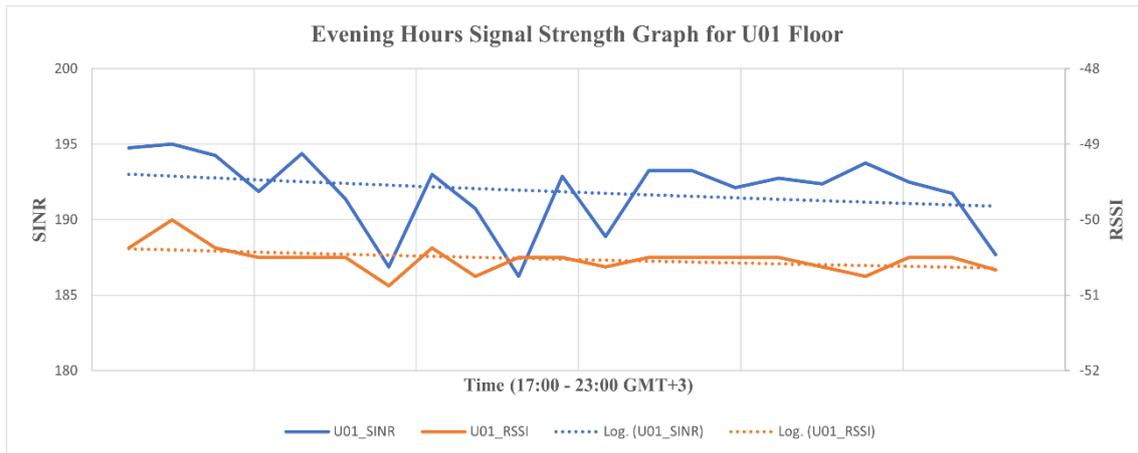


Figure 24: Signal strength graph for the first floor during evening hours

In the result for the first floor shown in section 4.1, the legends SINR_1/RSSI_1 and SINR_2/RSSI_2, in Figures 21 and 23, correspond to devices with IDs BG96_1 and BG96_2, respectively. The device with ID BG96_1 shows an excellent RSSI and SINR levels that remained stable throughout the morning hours (see Fig. 21), while a similar situation is observed in the evening hours with only slightly decreasing RSSI towards the end of the evening hours (see Fig. 23). The device with ID BG96_2 shows a decreasing RSSI and slightly increasing SINR values in the morning hours (see Fig. 21) while showing a slightly decreasing SINR and stable RSSI values in the evening hours (see Fig. 23).

Conclusively, the morning hours for the first floor shows the SINR and RSSI values slightly increasing and decreasing, respectively (see Fig. 22). The evening hours shows a gradually decreasing SINR and RSSI values for the first floor (see Fig. 24). The first floor generally shows an excellent coverage performance for all devices on the floor in the morning and evening hours.

4.2 Second Floor

The Figures 25 to 28 show the result of the measurement campaign on the second floor. These results represent the signal strength performance for each UE node device on the second floor and the general signal strength for all the devices on the second floor.

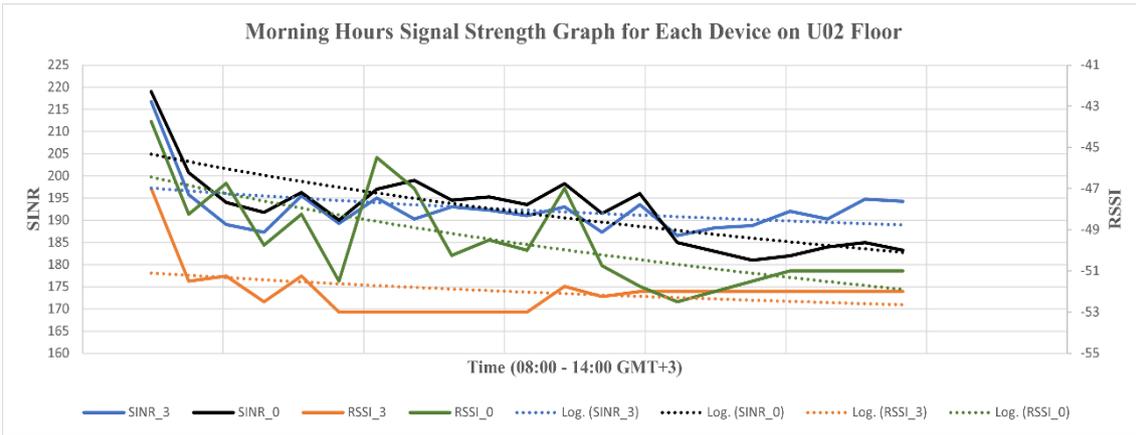


Figure 25: Signal strength graph for each UE node device on the second floor during morning hours

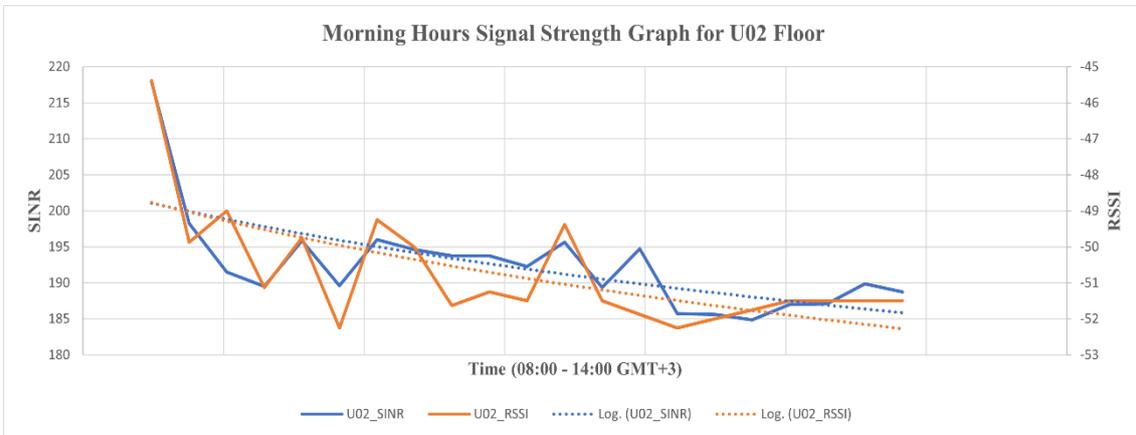


Figure 26: Signal strength graph for the second floor during morning hours

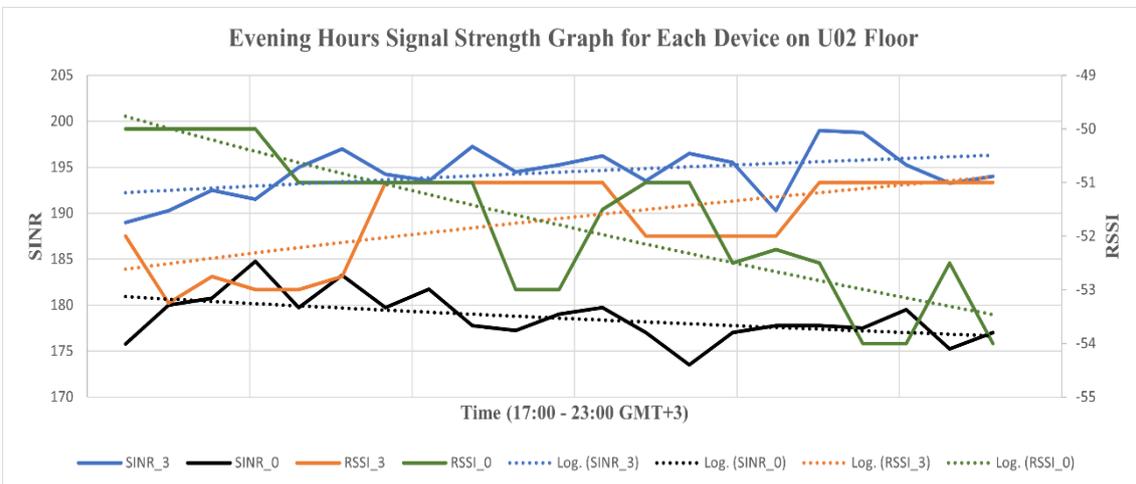


Figure 27: Signal strength graph for each UE node device on the second floor during evening hours

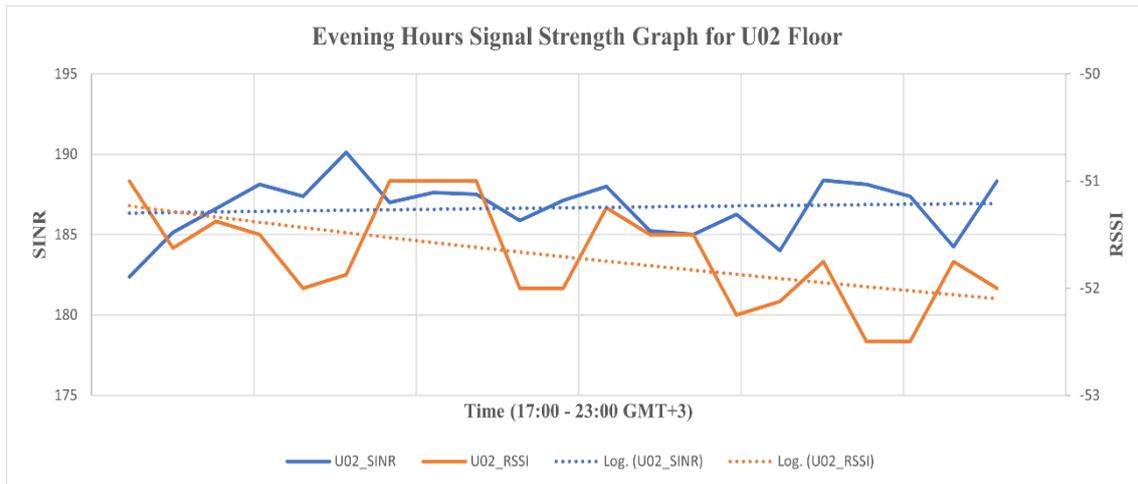


Figure 28: Signal strength graph for the second floor during evening hours

In the result for the second floor shown in section 4.2, the legends SINR_3/RSSI_3 and SINR_0/RSSI_0, in Figures 25 and 27, correspond to devices with IDs BG96_3 and BG96_0, respectively. The device with ID BG96_3 shows slightly decreasing SINR and RSSI values in the morning hours (see Fig. 25) while showing a slightly increasing SINR and RSSI values in the evening hours (see Fig. 27) that remained in the excellent performance level. The device with ID BG96_0 shows a decreasing SINR and RSSI values in the morning hours (see Fig. 25) while showing decreasing SINR and RSSI values in the evening hours (see Fig. 27).

Conclusively, the second floor shows a decreasing SINR and RSSI in the morning hours (see Fig. 26), while the evening hours shows a decreasing RSSI but stable SINR values (see Fig. 28). The second floor generally shows an excellent coverage performance for all devices on the floor in the morning and evening hours.

4.3 Third Floor

The Figures 29 to 32 show the result of the measurement campaign on the third floor. These results represent the signal strength performance for each UE node device on the third floor and the general signal strength for all the devices on the third floor.

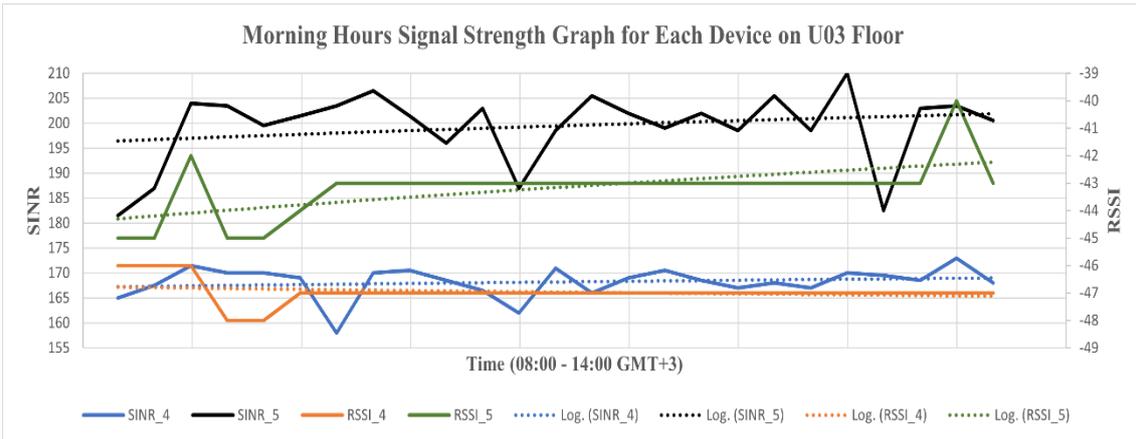


Figure 29. Signal strength graph for each UE node device on the third floor during morning hours

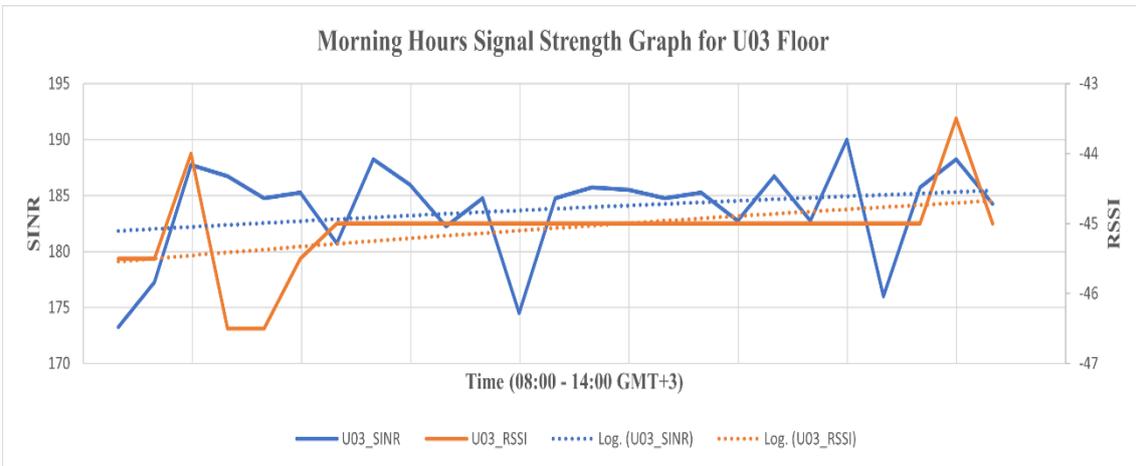


Figure 30: Signal strength graph for the third floor during morning hours

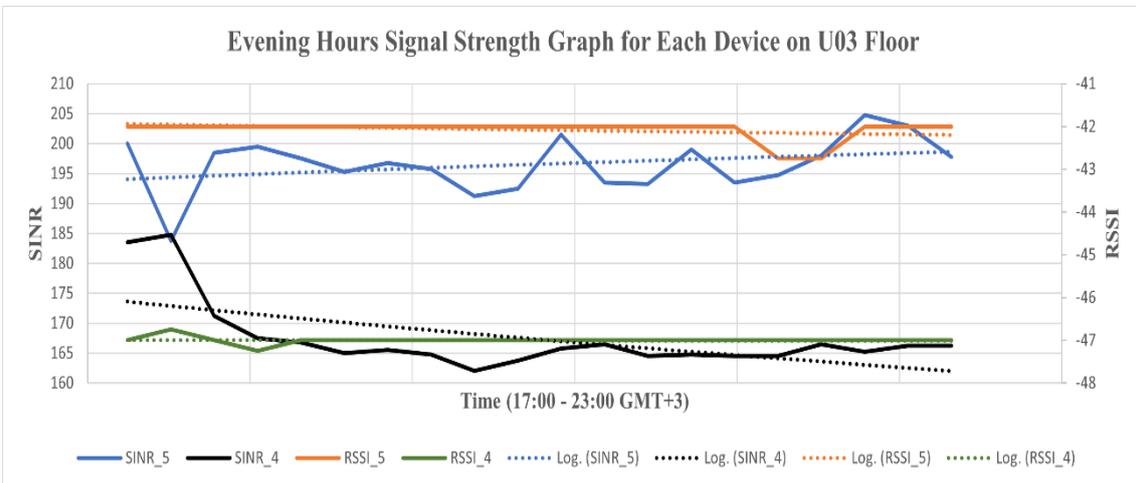


Figure 31: Signal strength graph for each UE node device on the third floor during evening hours

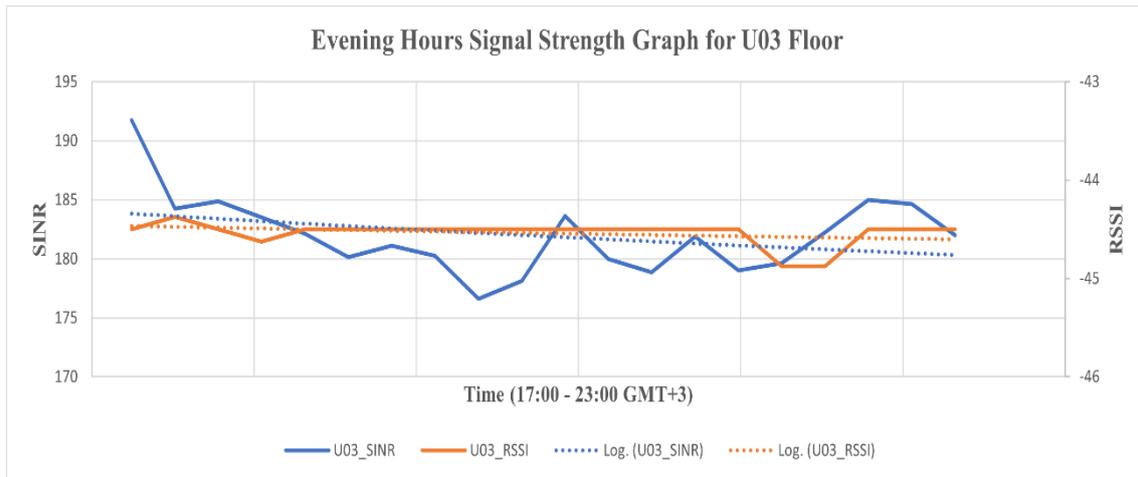


Figure 32: Signal strength graph for the third floor during evening hours

In the result for the third floor shown in section 4.3, the legends SINR_4/RSSI_4 and SINR_5/RSSI_5, in Figures 29 and 31, correspond to devices with IDs BG96_4 and BG96_5, respectively. The device with ID BG96_4 shows a stable SINR and RSSI values in the morning hours (see Fig. 29) while showing a stable RSSI and decreasing SINR values in the evening hours (see Fig. 31). The device with ID BG96_5 shows a slightly increasing SINR and RSSI in the morning hours (see Fig. 29) while showing a stable RSSI and slightly increasing SINR values in the evening hours (see Fig. 31).

Conclusively, the third floor shows slightly increasing SINR and RSSI values in the morning hours (see Fig. 30) while showing a stable RSSI and slightly decreasing SINR values in the evening hours (see Fig. 32). The third floor generally shows an excellent coverage performance for all devices on the floor in the morning and evening hours.

4.4 Fourth Floor

The Figures 33 to 36 show the result of the measurement campaign on the fourth floor. These results represent the signal strength performance for each UE node device on the fourth floor and the general signal strength for all the devices on the fourth floor.

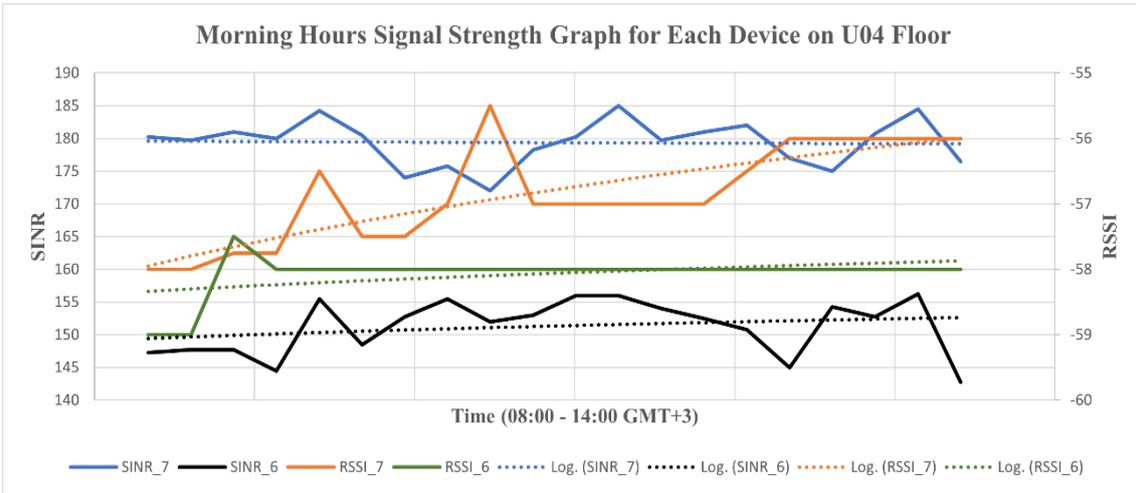


Figure 33: Signal strength graph for each UE node device on the fourth floor during morning hours

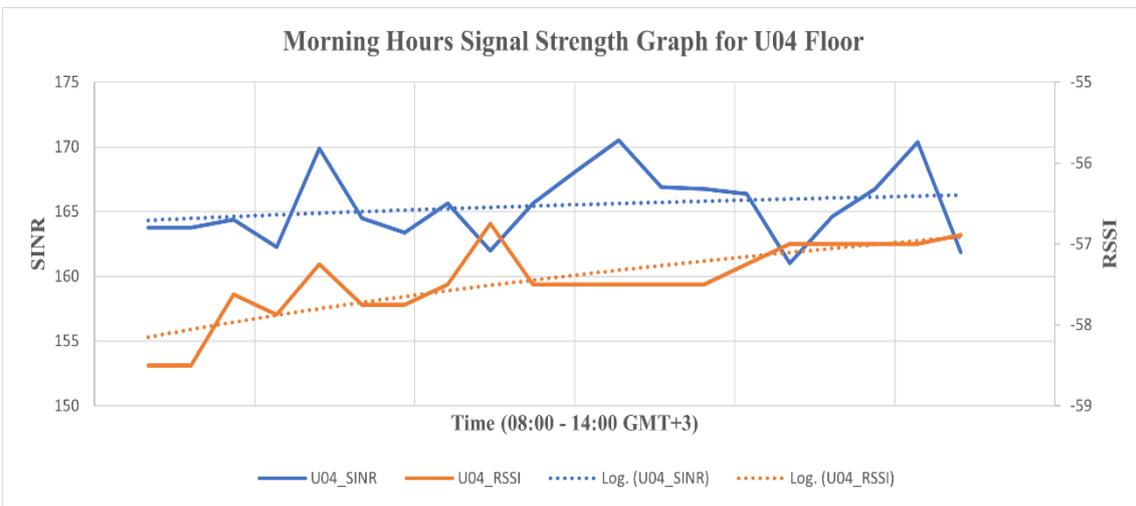


Figure 34: Signal strength graph for the fourth floor during morning hours

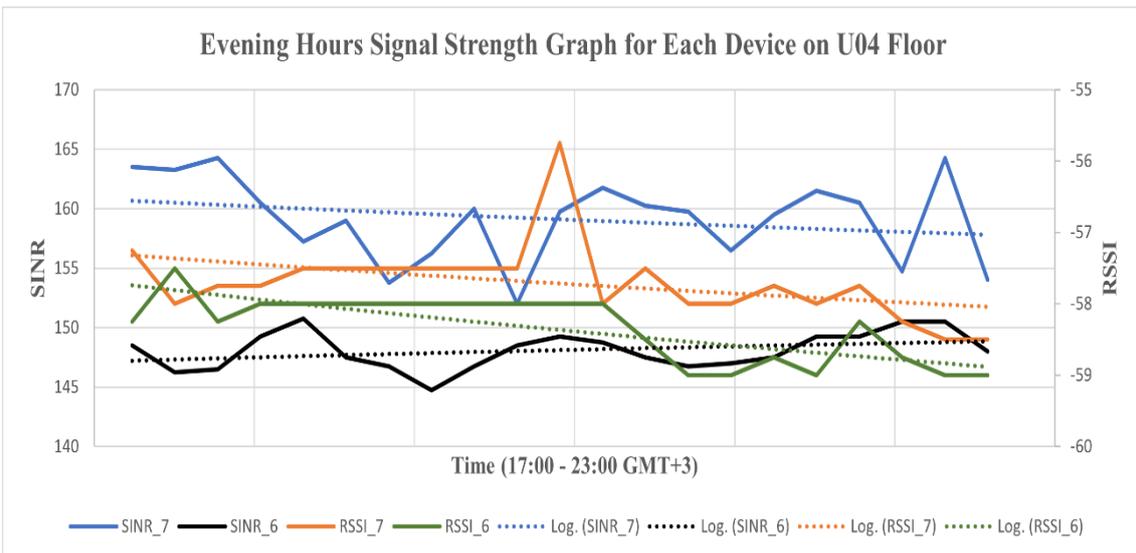


Figure 35: Signal strength graph for each UE node device on the fourth floor during evening hours

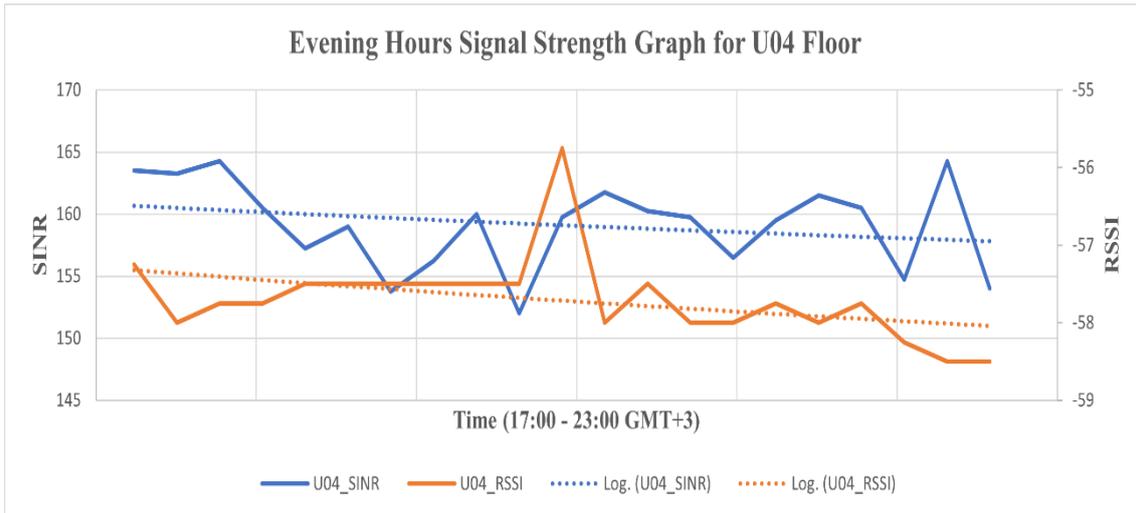


Figure 36: Signal strength graph for the fourth floor during evening hours

In the result for the fourth floor shown in section 4.4, the legends SINR_7/RSSI_7 and SINR_6/RSSI_6, in Figures 33 and 35, correspond to devices with IDs BG96_7 and BG96_6, respectively. The device with ID BG96_7 shows a stable SINR and increasing RSSI values in the morning hours (see Fig. 33) while showing slightly decreasing SINR and RSSI values in the evening hours (see Fig. 35). The device with ID BG96_6 shows slightly increasing SINR and RSSI values in the morning hours (see Fig. 33) while showing slightly increasing SINR and decreasing RSSI values in the evening hours (see Fig. 35).

Conclusively, the fourth floor shows an increasing SINR and RSSI values in the morning hours (see Fig. 34) while showing slightly decreasing SINR and RSSI values in the evening hours (see Fig. 36). The fourth floor generally shows an excellent coverage performance for all devices on the floor in the morning and evening hours.

The packet delivery ratio is calculated using the formula presented in this paper [56].

$$PDR = \frac{\sum Total\ number\ of\ packet\ received}{\sum Total\ number\ of\ packet\ sent}$$

Table 9: Average signal strength values for each floor

Floor Number	Morning hours (08:00 – 14:00 GMT+3)		Evening hours (17:00 – 23:00 GMT+3)	
	Avg. SINR (dB)	Avg. RSSI (dBm)	Avg. SINR (dB)	Avg. RSSI (dBm)
1	19	-50	18	-51
2	18	-51	17	-52
3	16	-45	16	-45
4	13	-57	11	-58

Table 10: Total packet delivery for each floor

Floor Number	Morning hours (08:00 – 14:00 GMT+3)			Evening hours (17:00 – 23:00 GMT+3)		
	Total packets sent	Total packets received	PDR	Total packets sent	Total packets received	PDR
1	234	234	1	325	325	1
2	234	234	1	324	324	1
3	312	312	1	303	303	1
4	290	290	1	324	324	1

As shown in Table 10, all the floors showed a strong signal coverage and 100% packet delivery ratio indicating that all packets sent by devices were received by the cloud system. The summary of signal strength and the packet delivery ratio values obtained for the field measurement campaign conducted on the four floors are shown in Tables 9 and 10, respectively.

5 Discussion

This chapter discusses the result presented in chapter 4 by analysing the results to understand the possible user experience and suggesting future improvement works.

5.1 Results Analysis

The results show some increases and decreases in the signal strength parameters for the various floor in the morning and evening hours. However, these parameters remained within the excellent performance value range based on Table 8, and all network packets sent by the devices were received in the cloud system. An overall stable signal performance can be observed on each floor for both morning and evening hours in most cases, although the results reveal some downward or upward slopes on the different floors within a small range of values. The downward and upward slopes in the graph imply signal degradation and improvement respectively, and the small range of values show the signal's stable performance. The observed slopes of the various graphs can be attributed to the dynamic environmental conditions of the air medium having various competing signals being transmitted together with several physical obstructions along the signal path. These limitations do not appear to impact the signal coverage as the values of the SINR and RSSI still fall within a high-performance range of 11 dB to 19 dB and -45 to -58 dBm, respectively, with a PDR of 1 for all devices, implying that IoT devices installed within this building area would have good quality network service.

Also, as shown in Table 9, the overall results show that the overall signal quality and power slightly reduces as the elevation level increases. This insight is opposite to the expected effect of altitude, where the signal quality should increase as the elevation level of the installed devices increase toward line-of-sight. This observation can be attributed to the peculiar environmental conditions of each floor. For example, the second floor has several fully staffed offices, furniture, etc., that obstruct signals travelling through the air and several equipped laboratories that are likely transmitting competing signals. The

gradual slight decrease in the overall signal values observed during the morning and evening hours on this floor can be ascribed to these conditions. As expected, the effect of these conditions is more evident in the morning hours than evening hours due to more human activities in the morning hours. Due to its small range, this decrease is not expected to impact user experience.

As predicted in section 2.4, the results obtained show a strong indoor coverage performance for LTE Cat-M1, which had been the case for NB-IoT in this paper [18]. However, an overall higher signal strength values were obtained in the thesis work. These higher results, which indicate a good quality network service, can be attributed to the proximity of the devices to the base station in this field measurement and the window installation of the physical device for less obstruction, *ceteris paribus*. Conclusively, IoT devices for various use-cases installed within this building area considered and connected to either the LTE Cat-M1 or NB-IoT network are expected to have quality service.

5.2 Future Work

Network coverage analysis is an activity that should happen periodically so that network providers can have updated state-of-the-network information to evaluate user experience. Although the measurement campaign location in this thesis is close to the base station, it would be helpful to conduct measurement campaigns for several days for larger areas including challenging locations such as underground basements and other deep indoor locations. Similar work is done this paper [24] for NB-IoT showing measurement campaigns conducted for several days in indoor, outdoor, and underground areas. Also, conducting field measurement campaigns in line-of-sight or near-line-of-sight conditions at different locations within the campus would help provide a further understanding of the expected service quality for devices installed in such conditions. Such measurements would help to give a broader insight into the performance of the LTE Cat-M1 network for the devices installed within a larger landscape.

This thesis work considers the packet delivery ratio; however, it would be helpful to investigate the latency and the UE's energy consumption during packet transmission in related future work. This consideration could help determine the level of support of the installed LTE Cat-M1 network for applications with strict latency requirements; also

determine the possibility of the UE to support up to 10 years of battery life as required (see Table 3).

Another consideration for related future works would be to scale up the number of devices and include a few non-stationary devices within the field measurement area to understand the possible changes in the network performance with an increased device density. Furthermore, including some actual use-cases such as smart-energy metering, with map-view visualization and live scenario testing of such use-cases would help provide insight into the network's coverage performance in supporting various live applications.

Finally, an extensive comparison between the deployed LTE Cat-M1 and NB-IoT networks within the same coverage area would be a valuable consideration for related future works. Comparison metrics could be their signal strength performance, data throughput for different use-cases, latency, UE's energy consumption during data transmission, PDR, etc. Such a comparison would provide a deep insight into the overall network performance for both networks. This insight is helpful to IoT network service providers for proper network optimization and management.

6 Summary

Forecasts mentioned in chapter 1 indicate that billions of IoT devices and network connections are expected to increase, therefore continuous network service performance analysis of the IoT communication network remains a crucial activity pertinent to provide support for various IoT use-cases and quality service delivery for a great user experience. Therefore, the primary goal of this thesis is to understand the network service performance of one of the IoT communication networks, LTE Cat-M1, within a coverage area by using the coverage analysis method.

In performing this analysis, the standards and objectives stipulated by the standardization body, 3GPP, were studied, and other related works previously done were first examined. As indicated in section 2.4, two approaches were identified: the eNB and the UE perspectives. The UE perspective was selected because of its proximity to the user and the possibility of applying actual use-cases to understand actual users' experience. Another reason is that the UE perspective approach has been successfully used in these similar studies [18] [24] in the department, although for NB-IoT technology, suggesting that following this approach presents the opportunity to leverage on the experiences from these studies and extend the learnings thereof. As this selected approach involves developing the end-to-end IoT system to investigate a similar KPI as in this previous work [18], subsequent considerations include the system design and communication protocol for data transfer.

The system's end-to-end design follows a modular design approach to allow for flexibility and faster design iterations. This proved especially helpful in the cloud node design part of this thesis work, which involved a complete pivot from the initial architecture due to the limitation in the packet capture details obtained and the complexity of the architecture. The final design addressed these limitations with a high level of simplicity. Also, the MQTT protocol was selected as the communication protocol for end-to-end IoT data transfer by this system ahead of other supported protocols like HTTP, TCP, FTP because of its suitability and prevalent industrial use. Similarly, UART serial communication

protocol was used at the UE node ahead of other supported protocols like I2C, SPI because of its simple implementation.

The measurement campaign in this thesis work was conducted at various altitudes to understand the different signal conditions and users' experience at different elevation levels. The results obtained indicate that users located within the area of TalTech main building, where this experiment was conducted, and connected to the LTE Cat-M1 network, would generally experience quality network service during the morning and evening hours regardless of their elevation level. However, those on the second floor may experience a relatively lesser signal quality during the morning hours, which could be attributed to other competing signals transmitted on the floor during these hours.

Finally, there are several improvement areas in this thesis work even though its primary goal was achieved, and some of these areas are suggested in section 5.3.

References

- [1] Mary Clare Novak, "A Brief History of Communication and Innovations that Changed the Game," G2, 04 April 2019. [Online]. Available: <https://www.g2.com/articles/history-of-communication>. [Accessed 08 May 2021].
- [2] Erik Josefsson, "How to improve ROI for Industry 4.0 use cases," Ericsson, 7 July 2020. [Online]. Available: <https://www.ericsson.com/en/blog/2020/7/how-to-improve-roi-for-industry-4-0-use-cases>. [Accessed 9 February 2021].
- [3] Beau Bass and G. Press, "New GSMA Study: Operators Must Look Beyond Connectivity to Increase Share of \$1.1 Trillion IoT Revenue Opportunity," GSM Association, 30 May 2018. [Online]. Available: <https://www.gsma.com/newsroom/press-release/new-gsma-study-operators-must-look-beyond-connectivity-to-increase-share/>. [Accessed 8 February 2021].
- [4] Fredrik Dahlgqvist, Mark Patel, Alexander Rajko and J. Shulman, "Growing opportunities in the Internet of Things," McKinsey & Company, 22 July 2019. [Online]. Available: <https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things>. [Accessed 8 February 2021].
- [5] GSMA Intelligence, "Beyond Connectivity: IoT Market Opportunities," GSM Association, [Online]. Available: <https://www.gsma.com/iot/beyond-connectivity-iot-market-opportunities/>. [Accessed 8 February 2021].
- [6] i-SCOOP, "LPWA network technologies and standards: LPWAN wireless IoT guide," i-SCOOP, [Online]. Available: <https://www.i-scoop.eu/internet-of-things-guide/lpwan/>. [Accessed 8 February 2021].
- [7] Hanish Bhatia, "LPWANs Will Co-Exist, No War Brewing Between Cellular and Non-Cellular," Counterpoint Technology Market Research, 9 August 2018. [Online]. Available: <https://www.counterpointresearch.com/lpwans-will-co-exist-no-war-brewing-between-cellular-non-cellular/>. [Accessed 8 February 2021].
- [8] Kais Mekki, Eddy Bajic, Frederic Chaxel and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," 20 December 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405959517302953>. [Accessed 16 02 2021].
- [9] GSMA, "LTE-M Deployment Guide to Basic Feature Set Requirements," June 2019. [Online]. Available: <https://www.gsma.com/iot/wp-content/uploads/2019/08/201906-GSMA-LTE-M-Deployment-Guide-v3.pdf>. [Accessed 10 February 2021].
- [10] 3. G. P. P. 3GPP, "3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT) (Release 13)," 2015.

- [11] TalTech, "A CLIMATE SMART TALTECH BY 2035," TalTech, 03 February 2021. [Online]. Available: <https://taltech.ee/en/news/video-climate-smart-taltech-2035>. [Accessed 15 February 2021].
- [12] Altair Semiconductor Limited, "Coverage Analysis of LTE-M Category-M1, White Paper," January 2017. [Online]. Available: <https://altair-semi.com/wp-content/uploads/2017/02/Coverage-Analysis-of-LTE-CAT-M1-White-Paper.pdf>. [Accessed 12 February 2021].
- [13] Keysight Technologies Inc, "LTE Physical Layer Overview," Keysight Technologies, Inc, 2000-2021. [Online]. Available: http://rfmw.em.keysight.com/wireless/helpfiles/89600b/webhelp/subsystems/lte/content/lte_overview.htm. [Accessed 4 March 2021].
- [14] Olof Liberg, Y.-P. Eric Wang, Joachim Sachs, Mårten Sundberg, Johan Bergman and G. Wikström, "LTE-M," in *Cellular Internet of Things*, Academic Press, 2019, pp. 155-254.
- [15] Agilent Technologies Inc, "Agilent 3GPP Long Term Evolution: System Overview, Product Development, and Test Challenges," 8 September 2009. [Online]. Available: <https://literature.cdn.keysight.com/litweb/pdf/5989-8139EN.pdf?id=1431418>. [Accessed 3 April 2021].
- [16] 3. G. P. P. 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on provision of low-cost Machine-Type Communications (MTC) User Equipments (UEs) based on LTE (Release 12)," June 2013. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/36_series/36.888?sortby=size. [Accessed 27 January 2021].
- [17] Rohde & Schwarz, "UMTS Long Term Evolution (LTE) Technology Introduction," 23 October 2012. [Online]. Available: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ma111/1MA111_4E_LTE_technology_introduction.pdf. [Accessed 1 March 2021].
- [18] Sikandar Zulqarnain Khan, Hassan Malik, Jeffrey Leonel Redondo Sarmiento, Muhammad Mahtab Alam and Y. L. Moullec, "DORM: Narrowband IoT Development Platform and Indoor Deployment Coverage Analysis," *Procedia Computer Science, The 2nd International Workshop on Recent Advances in Cellular Technologies and 5G for IoT Environments (RACT-5G-IoT 2019)*, vol. 151, pp. 1084-1091, 2019.
- [19] Mads Lauridsen, Istvan Z. Kovacs, Preben Mogensen, Mads Sorensen and Steffen Holst, "Coverage and Capacity Analysis of LTE-M and NB-IoT in a Rural Area," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Montreal, QC, Canada, 2016.
- [20] István Z. Kovács, Preben Mogensen, Mads Lauridsen, Thomas Jacobsen, Krzysztof Bakowski, Poul Larsen, Nitin Mangalvedh and Rapeepat Ratasuk, "LTE IoT link budget and coverage performance in practical deployments," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, QC, Canada, 2017.
- [21] Gus Vos, Johan Bergman, Yigal Bitran, Martin Beale, Mark Cannon, Robert Holden, Yee Sin Chan, Ron Toledano, Ronan Le Bras, Toshiyasu Wakayama, Rapeepat Ratasuk, Naoto Okubo, Kyujin Park, Yosuke Akimoto and S. L. Tronic Haholongan Siregar, "Coverage Analysis of LTE-M, Category-M1," January 2017.

- overview/#:~:text=The%20maximum%20packet%20size%20is,etc.. [Accessed 19 April 2021].
- [34] Steve Cope, "MQTT Brokers and Cloud Hosting Guide," Steve's internet Guide, 2011. [Online]. Available: <http://www.steves-internet-guide.com/mqtt-hosting-brokers-and-servers/>. [Accessed 22 April 2021].
- [35] The HiveMQ Team, "Getting Started with MQTT," HiveMQ GmbH, 24 April 2020. [Online]. Available: <https://www.hivemq.com/blog/how-to-get-started-with-mqtt/>. [Accessed 20 April 2021].
- [36] Pamela Fox, "Transmission Control Protocol (TCP)," Khan Academy, 2021. [Online]. Available: <https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:transporting-packets/a/transmission-control-protocol--tcp>. [Accessed 20 April 2021].
- [37] Henrik Frystyk, "The Internet Protocol Stack," The Internet Protocol Stack, July 1994. [Online]. Available: <https://www.w3.org/People/Frystyk/thesis/TcpIp.html#TCP>. [Accessed 20 April 2021].
- [38] Cisco Certified Expert, "Positive Acknowledgment and Retransmission," Cisco Certified Expert, 06 January 2021. [Online]. Available: <https://www.ccxpert.us/network-layer/positive-acknowledgment-and-retransmission.html>. [Accessed 20 April 2021].
- [39] Digital Guide Ionos, "TCP (Transmission Control Protocol) – The transmission protocol explained," 1&1, 2 March 2020. [Online]. Available: <https://www.ionos.com/digitalguide/server/know-how/introduction-to-tcp/>. [Accessed 20 April 2021].
- [40] RandomNerdTutorials, "What is MQTT and How It Works," RandomNerdTutorials, 2013. [Online]. Available: <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>. [Accessed 16 April 2021].
- [41] Steve Cope, "Beginners Guide To The MQTT Protocol," Steve's internet Guide, 2011. [Online]. Available: <http://www.steves-internet-guide.com/mqtt/>. [Accessed 20 April 2021].
- [42] MQTT, "MQTT: The Standard for IoT Messaging," MQTT, 2020. [Online]. Available: <https://mqtt.org/>. [Accessed 19 April 2021].
- [43] OpenLab Pro, "MQTT Packet Format," Etiq Technologies, 2019. [Online]. Available: <https://openlabpro.com/guide/mqtt-packet-format/>. [Accessed 19 April 2021].
- [44] Eric Peña and Mary Grace Legaspi, "UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter," Analog Devices, December 2020. [Online]. Available: <https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>. [Accessed 21 April 2021].
- [45] Scott Campbell, "BASICS OF UART COMMUNICATION," Circuit Basics, [Online]. Available: <https://www.circuitbasics.com/basics-uart-communication/>. [Accessed 19 February 2021].
- [46] OpenMV LLC, "Download, OpenMV IDE v2.6.9," OpenMV LLC, 13 April 2021. [Online]. Available: <https://openmv.io/pages/download>. [Accessed 21 April 2021].
- [47] Damien P. George, Paul Sokolovsky and OpenMV LLC, "4. OpenMV IDE Overview," Micropython, OpenMV, 28 January 2021. [Online]. Available:

- https://docs.openmv.io/openmvcam/tutorial/openmvide_overview.html. [Accessed 21 April 2021].
- [48] Walter LI and Harvey HE, "QCOM User Guide - Quectel Forums," 20 03 2015. [Online]. Available: <https://forums.quectel.com/uploads/short-url/y1LgPXaeeKRvCUPILhvRz6cp6aR.pdf>. [Accessed 21 April 2021].
- [49] PuTTYgen, "Download PuTTY for Windows , Linux and Mac," PuTTYgen, [Online]. Available: <https://www.puttygen.com/download-putty>. [Accessed 21 April 2021].
- [50] SolarWinds Worldwide LLC, "Solar-PuTTY," SolarWinds Worldwide, LLC, 2021. [Online]. Available: <https://www.solarwinds.com/free-tools/solar-putty>. [Accessed 26 April 2021].
- [51] Wireshark, "Wireshark User's Guide," Wireshark, [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/index.html. [Accessed 21 April 2021].
- [52] Amazon Web Services, "What is the AWS Management Console?," Amazon Web Services, [Online]. Available: <https://docs.aws.amazon.com/awsconsolehelpdocs/latest/gsg/learn-whats-new.html>. [Accessed 21 April 2021].
- [53] Amazon Web Services, "Amazon Management Console," Amazon Web Services, [Online]. Available: <https://aws.amazon.com/console/>. [Accessed 21 April 2021].
- [54] Digi International Inc., "Example: receive messages (subscribe) with MQTT," 05 November 2020. [Online]. Available: https://www.digi.com/resources/documentation/Digidocs/90001541/reference/r_example_subscribe_mqtt.htm. [Accessed 23 April 2021].
- [55] Johannes 4GNU_Linux, "MQTT Clients in Python with the paho-mqtt module," YouTube, 14 September 2020. [Online]. Available: https://www.youtube.com/watch?v=c_DPKujOmGw&t=3s. [Accessed 16 April 2021].
- [56] Ali H. Wheeb, Ameer Morad and Maad Issa Al-Tameemi, "Performance Evaluation of Transport Protocols for Mobile Ad Hoc Networks," *Journal of Engineering and Applied Sciences*, vol. 13, no. 13, pp. 5181-5185, 2018.

Appendix 1 – Non-exclusive Licence for Reproduction and Publication of a Graduation Thesis¹

I Stanley Ohumegbulem Osajeh

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis CAT-M1 Coverage Analysis for Demo Campus Environment, supervised by Muhammad Mahtab Alam and Sven Päränd
 - 1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
 - 1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

10.05.2021

¹ The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – MQTT Subscriber Python Script

```
import paho.mqtt.client as iot
import sys
import json
import csv

def csvconversion(data):
    rawdata = json.loads(data)
    fields = ['tx_bytes','rx_bytes','sinr','rssi', 'start_time', 'finish_time', 'image_size']
    with open('u02_22-04-2021.csv', 'a+', newline = "") as csvdata:
        writer = csv.DictWriter(csvdata, fieldnames = fields)
        writer.writerow(rawdata)
        csvdata.close()

def my_message(client, userdata, msg):
    data_in = msg.payload.decode()
    data_conv = data_in
    csvconversion(data_conv)
    print('done')

client = iot.Client()
client.on_message = my_message

if client.connect("13.53.216.46", 1883, 60) != 0:
    print("Connected could not be established to the MQTT broker!")
    sys.exit(-1)

client.subscribe("thesis/data/u01/device")

try:
    print("Press CTRL+C to exit")
    client.loop_forever()
except:
    print("Disconnecting this client from broker")

client.disconnect()
```

Appendix 3 – UE Node Program Code

```
import time, utime, machine, pyb, sensor, ujson, ubinascii, image, uos
from pyb import UART

uart = UART(3, 115200, timeout_char=1000)
uart.init(115200, bits=8, parity=None, stop=1, timeout=1500, timeout_char=1500)

RED_LED_PIN = 1
BLUE_LED_PIN = 3

def readin():
    output = uart.read()
    return output.decode('utf-8')

def send_data():
    if uart.any(): readin()
    piece_size = 1024
    with open("pic1.jpg", 'rb') as my_image:

        while True:
            data_piece = my_image.read(piece_size)
            data = ubinascii.b2a_base64(data_piece)
            data_json = ujson.dumps(data)
            if not data_piece: break

            # sending image data and collect timestamps
            uart.write("AT+CCLK?\r\n")
            starttime = readin()
            uart.write('AT+QMTOPEN=2,"13.53.216.46",1883\r\n')
            pyb.delay(10)
            readin()
            uart.write('AT+QMTCONN=2,"bg96_0"\r\n')
            pyb.delay(10)
            readin()
            uart.write('AT+QMTCPUB=2,0,0,0,"thesis/data/u01/cloud"\r\n')
            pyb.delay(5)
            uart.write(data_json)
            pyb.delay(3)
            uart.write(b'\x1A\r\n')
            pyb.delay(10)
            readin()
            uart.write("AT+CCLK?\r\n")
            finishtime = readin()

            starttime = starttime.split()
            finishtime = finishtime.split()
```

```

timedata = [starttime[2], finishtime[2]]
return timedata

def trx_details(devicedata_json):
    #sending data collected on the end-node side
    uart.write('AT+QMTOPEN=2,"13.53.216.46",1883\r\n')
    pyb.delay(10)
    readin()
    uart.write('AT+QMTCONN=2,"bg96_0"\r\n')
    pyb.delay(10)
    readin()
    uart.write('AT+QMT PUB=2,0,0,0,"thesis/data/u01/device"\r\n')
    pyb.delay(5)
    uart.write(devicedata_json)
    pyb.delay(3)
    uart.write(b'\x1A\r\n')

def capture_image():
    # Capture picture data by taking a snapshot
    sensor.reset() # Initialize the camera sensor.
    sensor.set_pixformat(sensor.GRAYSCALE) # or sensor.GRAYSCALE
    sensor.set_framesize(sensor.QQVGA) # or sensor.QQVGA (or others)
    sensor.skip_frames(time = 2000) # Let new settings take affect.

    pyb.LED(RED_LED_PIN).on()
    sensor.skip_frames(time = 2000) # Give the user time to get ready.

    pyb.LED(RED_LED_PIN).off()
    pyb.LED(BLUE_LED_PIN).on()

    filesize = str(uos.stat('pic1.jpg'))
    filesize = filesize.split(",")
    img_size = filesize[6]

    pyb.LED(BLUE_LED_PIN).off()
    print("Done! Reset the camera to see the saved image.")
    return img_size

def initialization():
    # Initialize connection with BG96 module and set it up for sending data
    print("program is running")
    uart.write("ATI\r\n")
    readin()
    uart.write("AT+CPIN=0000\r\n")
    readin()
    uart.write("AT+CREG=2\r\n")
    pyb.delay(30)
    uart.write("AT+CREG?\r\n")
    readin()
    uart.write("AT+CEREG=2\r\n")

```

```

pyb.delay(10)
uart.write("AT+CEREG?\r\n")
readin()
uart.write("AT+COPS=0,0,'24801',8\r\n")
uart.write("AT+COPS?\r\n")
readin()
uart.write('AT+QCFG="iotopmode",2\r\n')
uart.write('AT+QCFG="iotopmode"\r\n')
readin()
uart.write('AT+QCFG="nwscanmode",3\r\n')
uart.write('AT+QCFG="nwscanmode"\r\n')
readin()
uart.write('AT+QCFG="band",f,400a0e189f,a0e189f\r\n')
uart.write('AT+QCFG="band"\r\n')
readin()
uart.write('AT+QCFG="nwscanseq",00,1\r\n')
readin()
uart.write('AT+QCFG="nwscanseq"\r\n')
readin()

uart.write('AT+QICSGP=2,1,"internet.emt.ee", "", "", 0\r\n')
pyb.delay(10)
uart.write("AT+QICSGP=2\r\n")
readin()
uart.write("AT+QIACT=2\r\n")
pyb.delay(10)
uart.write("AT+QIACT?\r\n")
readin()
uart.write("AT+CTZU=1\r\n")
readin()
uart.write('AT+QNTP=2,"ee.pool.ntp.org"\r\n')
pyb.delay(10)
readin()

```

```

#####
# MAIN PROGRAM #
#####

```

```

# indicate start program running
pyb.LED(RED_LED_PIN).on()
time.sleep(5)
pyb.LED(RED_LED_PIN).off()
time.sleep(5)

```

```

# initialize BG96
initialization()

```

```

# take a snapshot
img_size = capture_image()

```

```
while (True):
```

```
    # indicate start program running
    pyb.LED(RED_LED_PIN).on()
    time.sleep(10)
    pyb.LED(RED_LED_PIN).off()
    time.sleep(5)
```

```
    # clear BG96 packet data counter
    uart.write("AT+QGDCNT=0\r\n")
    pyb.delay(10)
    readin()
```

```
    # collect signal parameter
    uart.write("AT+QCSQ\r\n")
    pyb.delay(10)
    sig_data = readin()
    sig_data = sig_data.split()
    sig = sig_data[2].split(",")
```

```
    # send image to AWS cloud
    timespan = send_data()
    pyb.delay(5)
```

```
    # collect data on end-node side
    uart.write("AT+QGDCNT?\r\n")
    pyb.delay(20)
    pkt_info = readin()
    pkt_info = pkt_info.split()
    pkt = pkt_info[2].split(",")
    devicedata =
```

```
{ "tx_bytes":pkt[0], "rx_bytes":pkt[1], "sinr":sig[3], "rssi":sig[1], "start_time":timespan[0],
  "finish_time":timespan[1], "image_size":img_size }
```

```
    devicedata_json = ujson.dumps(devicedata)
```

```
    # send to AWS cloud
    trx_details(devicedata_json)
```

```
    # close connection
    uart.write('AT+QMTDISC=2\r\n')
    pyb.delay(10)
    readin()
```

```
    print("OPERATION COMPLETED")
```

```
    # wait for 5 min
    utime.sleep(300)
```