

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2023

Patrik Horčíčka



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

INTELIGENTNÍ MĚŘENÍ ENERGIÍ PRO SVJ S VYUŽITÍM 5G

INTELLIGENT ENERGY MEASUREMENTS FOR HOMEOWNERS' ASSOCIATION USING 5G NETWORKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Patrik Horčíčka

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Mašek, Ph.D.

BRNO 2023

Bakalářská práce

bakalářský studijní program **Telekomunikační a informační systémy**

Ústav telekomunikací

Student: Patrik Horčíčka

ID: 230558

Ročník: 3

Akademický rok: 2022/23

NÁZEV TÉMATU:

Inteligentní měření energií pro SVJ s využitím 5G

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce bude v teoretické části nastudovat aktuálně dostupné metody pro měření energií v rámci SVJ, tj., společenství vlastníků jednotek. Pozornost bude soustředěna zejména na klíčové energie (elektřina, voda, plyn) a měřiče tepla (kalorimetry). V rámci praktické části bakalářské práce bude přistoupeno k vytvoření referenčního 5G-IoT zařízení pro komunikaci s vybranými měřiči energií a pro následný přenos dat k uživateli/dohledovému systému. Pro přenos dat budou využity 5G komunikační technologie v rámci Průmyslu 4.0, tj., Narrowband IoT či LTE Cat-M. Funkcionalita zařízení bude ověřena v reálných podmínkách ve spolupráci s operátorem Vodafone Česká republika.

DOPORUČENÁ LITERATURA:

[1] CHAUDHARI, Bharat S. a Marco ZENNARO, 2020. LPWAN Technologies for IoT and M2M Applications. Online: Academic Press.

[2] LIBERG, Olof, Mårten SUNDBERG, Y.-P. Eric WANG, Johan BERGMAN a Joachim SACHS, [2018]. Cellular Internet of things: technologies, standards, and performance. San Diego, CA, United States: Academic Press, an imprint of Elsevier. ISBN 978-012-8124-581.

Termín zadání:

Termín odevzdání: 17.8.2023

Vedoucí práce: Ing. Pavel Mašek, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

ÚPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRACT

Theoretical analysis of IoT communication technologies LPWAN. A more detailed analysis of 5G-IoT technologies in the licensed frequency band, such as NB-IoT and LTE Cat-M. Comparison of different energy meters (Smartmeters). List of supported application protocols for IoT devices (MQTT, CoAP, LwM2M, DLMS/COSEM) with more detailed information about DLMS/COSEM. Quectel UMTS & LTE EVB kit with BG770A-GL module was used to create the first reference 5G-IoT device. Measuring of radio properties in two locations. Creation of TLS tunnel for communication between server and RPI designed for DLMS/COSEM data.

KEYWORDS

IoT, H2H, M2M, SM, mMTC, 2G, 4G, 5G, CIoT, LPWAN, LoRaWAN, Sigfox, NB-IoT, LTE Cat-M, PSM, eDRX, 3GPP, MQTT, CoAP, LwM2M, DLMS/COSEM, IPSec, TLS, UMTS & LTE EVB Kit, BG770A, BG77, Raspberry Pi 4, DietPi, Python, Java, Gurex

ABSTRAKT

Teoretický rozbor IoT komunikačních technologií LPWAN. Detailnější analýza 5G-IoT technologií v licenčním frekvenčním pásmu, jako jsou NB-IoT a LTE Cat-M. Porovnání různých měřičů energií (Smartmetrů). Výčet podporovaných aplikačních protokolů u IoT zařízení (MQTT, CoAP, LwM2M, DLMS/COSEM) se detailnějším zaměřením na DLMS/COSEM. Pro vytvoření prvního referenčního 5G-IoT zařízení bylo využito Quectel UMTS & LTE EVB Kitu s BG770A-GL modulem. Finální zařízení kombinací RPi a BG77. Provedená měření rádiových vlastností v různých lokalitách. Vytvoření TLS tunelu pro komunikaci mezi server a RPi pro přenos DLMS/COSEM dat.

KLÍČOVÁ SLOVA

IoT, H2H, M2M, SM, mMTC, 2G, 4G, 5G, CIoT, LPWAN, LoRaWAN, Sigfox, NB-IoT, LTE Cat-M, PSM, eDRX, 3GPP, MQTT, CoAP, LwM2M, DLMS/COSEM, IPSec, TLS, UMTS & LTE EVB Kit, BG770A, BG77, Raspberry Pi 4, DietPi, Python, Java, Gurex

Author's Declaration

Author:	Patrik Horčíčka
Author's ID:	230558
Paper type:	Bachelor's Thesis
Academic year:	2022/23
Topic:	Inteligentní měření energií pro SVJ s využitím 5G

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno
author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I want to thank my thesis advisor, Ing. Pavel Mašek, PhD. for his valuable comments, sharing his knowledge and guidance of my thesis. I would also like to thank Ing. Radek Možný for his practical advice on the hardware part of the bachelor thesis. A special thanks to Jakub for all his advice in finalizing phase of the thesis. Last but not least, I would like to thank Alex for all her support.

Contents

Introduction	17
1 IoT (Internet of Things)	19
1.1 Smart Meters	19
1.1.1 Smart Gas Meter	20
1.1.2 Smart Water Meter	20
1.1.3 Smart Calorimeter	20
1.1.4 Smart Electric Meter	21
1.2 Communication Principles	21
1.2.1 H2H (Human-to-Human)	21
1.2.2 M2M (Machine-to-Machine)	21
1.2.3 mMTC (massive Machine-Type Communication)	21
1.3 Legacy Communication Technologies	22
1.3.1 The Second Generation	22
1.3.2 The Fourth Generation	22
1.4 LPWAN (Low-Power Wide Area Networks)	23
1.5 Communication Technologies	23
1.5.1 Sigfox	23
1.5.2 LoRaWAN	24
1.5.3 NB-IoT	26
1.5.4 LTE Cat-M	27
1.6 Comparison of Communication Technologies	28
2 CIoT (Cellular Internet of Things)	29
2.1 3GPP (3rd Generation Partnership Project)	29
2.2 Low Battery Consumption Modes	30
2.2.1 PSM (Power Saving Mode)	30
2.2.2 eDRX (extended Discontinuous Reception)	30
2.3 NB-IoT (Narrow-Band Internet of Things)	31
2.3.1 Modes of operation	31
2.3.2 Communication Channels	32
2.4 LTE Cat-M	33
2.4.1 Transmission Schemes	33
2.4.2 Communication Channels	34

3	Supported Application and Security Protocols	39
3.1	MQTT (Message Queuing Telemetry Transport)	39
3.2	CoAP (Constrained Application Protocol)	39
3.3	LwM2M (Lightweight Machine-to-Machine)	40
3.4	IPSec (Internet Protocol Security)	41
3.5	TLS (Transport Layer Security)	41
3.6	DLMS/COSEM (Device Language Message Specification/Companion Specification for Energy Metering)	42
3.6.1	DLMS/COSEM Three-Step Approach	42
3.6.2	DLMS/COSEM Objects	43
3.6.3	DLMS/COSEM Application Layer	44
3.6.4	DLMS/COSEM Messaging	45
3.6.5	DLMS/COSEM Communication Profiles	46
3.6.6	AAs (Application Associations)	47
3.6.7	DLMS/COSEM Security	48
4	Practical Part	51
4.1	Devices Used for Communication	51
4.1.1	QUECTEL LPWA BG77 Cat M1/NB2	51
4.1.2	RPI CM4 (Raspberry Pi Compute Module 4 Lite)	51
4.2	AT Commands	52
4.3	Radio Properties for Final Communication	55
4.3.1	Signal Properties	56
4.3.2	Measured Speeds and Sizes	57
4.3.3	Measured Delays	62
4.4	Infrastructure of DLMS/COSEM Communication	65
4.5	TLS Communication Tunnel	66
4.6	Communication	67
4.6.1	Client Side of the Communication	67
4.6.2	Server Side of the Communication (RPi)	67
	Conclusion	71
	Bibliography	73
	Symbols and abbreviations	77
5	Examples of Used Python Codes	81
6	Content of the electronic attachment	83

List of Figures

1.1	3GPP Technologies Evolution	23
1.2	Sigfox Communication Infrastructure	24
1.3	Sigfox Coverage of Czech Republic	24
1.4	LoRaWAN Communication Infrastructure	25
1.5	LoRaWAN Coverage of the Whole World	25
1.6	NB-IoT Communication Infrastructure	26
1.7	Vodafone NB-IoT Coverage of Czech Republic	26
1.8	LTE-Cat-M Communication Infrastructure	27
1.9	LTE-Cat-M Coverage of the Whole World	27
2.1	Summary Diagram eDRX	30
2.2	NB-IoT Modes of Operation	31
3.1	MQTT Communication of Client and Server	39
3.2	CoAP Communication of Client and Server	40
3.3	LWM2M Registration and Communication of Client and Server	40
3.4	DLMS/COSEM Communication Example	42
3.5	DLMS/COSEM Class and Object Examples	43
3.6	The Structure of the DLMS/COSEM Application Layers	44
3.7	DLMS/COSEM on ISO/OSI Model	45
3.8	DLMS/COSEM Messaging Patterns	46
3.9	DLMS/COSEM Communication Profile	47
4.1	QUECTEL LPWA BG77 Cat M1/NB2	52
4.2	5G laboratory special desk	52
4.3	Raspberry Pi Compute Module 4 Lite	53
4.4	AT-Command Flowchart	54
4.5	The Dependence of Sender Data Size on Sending Time in Brno	58
4.6	The Dependence of Receiver Data Size on Sending Time in Brno	59
4.7	The Dependence of Sender Bitrate on Sending Time in Brno	59
4.8	The Dependence of Receiver Bitrate on Sending Time in Brno	60
4.9	The Dependence of Sender Data Size on Sending Time in Heřmanice	61
4.10	The Dependence of Receiver Data Size on Sending Time in Heřmanice	61
4.11	The Dependence of Sender Bitrate on Sending Time in Heřmanice	62
4.12	The Dependence of Receiver Bitrate on Sending Time in Heřmanice	62
4.13	The Dependence of Packet Loss on Packet Size for Brno and Heřmanice	63
4.14	Measured Values With Ping for Different Package Sizes in Brno	64
4.15	Measured Values With Ping for Different Package Sizes in Heřmanice	65
4.16	Real Infrastructure of Connection	65

4.17 Client Console Output	69
4.18 Server Console Output	70

List of Tables

1.1	Comparison of Communication Technologies	28
4.1	NB-IoT Boundary Condition	55
4.2	Signal Properties in Brno	56
4.3	Signal Properties in Heřmanice	57

Introduction

With increasing prices of all energy sources, like electricity and gas, there is a need to control this consumption. People increasingly want to see their real-time consumption, so there is an upswing of smart meters in electrical measuring. Thanks to smart meters, there can be online monitoring for consumption and production in case of an installed photovoltaic system. In one part, there is more information about energy smart meters. However, now, the need for monitoring struggles with meters that do not have the option of online communication.

That is why LPWAN (Low Power Wide Area Network) technologies were introduced in the last years, which have better communication coverage and lower demands on the strength of the radio signals so the meters in cellars or other bad environments for signals can still communicate. This thesis provides more information about LPWAN technologies in licensed frequency bands like NB-IoT (Narrow-Band Internet of Things) and LTE Cat-M.

For metering purposes can be used several communication protocols. At the end of the theoretical part, there is more information about chosen protocols, like MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), LwM2M (Lightweight Machine-to-Machine), and DLMS/COSEM (Device Language Message Specification/Companion Specification for Energy Metering). Especially about DLMS/COSEM, there are more details and information about it because it will be chosen for final communication purposes.

In the first third of the practical part of this thesis, there is more information about the used device and how its main parts separately work.

In the second third of the practical part, the radio properties are measured in different locations for used NB-IoT devices. This part also contains information about the importance of individual radio properties for NB-IoT communication.

The final third of the practical part is information about infrastructure and communication with the final device. Also, there is a detailed description of each communication side and examples of console output.

1 IoT (Internet of Things)

It is a way that modern technological society changes. In the past, most objects used single-purpose communication. This is way too much work to connect more devices for automatization. The IoT provides the opportunity to simplify the automatization process. IoT systems connect more devices to one network, which can control actions and get sensing from the physical asset of the IoT device. A typical IoT endpoint is a simple, low-power embedded device emphasizing low cost and power consumption for battery-power devices. The more IoT devices we connect to the network more information we have available. Therefore, we can achieve even more optimization and automatization[1].

The fastest growing type of IoT is IIoT (Industry Internet of Things), mainly because modern companies are trying to optimize their process with the application of Industry 4.0. Furthermore, one of the main features of Industry 4.0 is IoT and other automatization. Another type of IoT is CIoT (Consumer Internet of Things), where we use intelligent devices. For example, smart devices for consumers are innovative home products like smart electricity meters, smart thermostats, or wearables like smartwatches or sports trackers. Another type of IoT can be IoMT (Internet of Medical Things). In medicine, we can use smart devices to improve patients' life or at least make it safer for them. There are remote health monitoring and emergency notification systems. These systems can monitor blood pressure, heart rates, or other important properties of the human body[2, 3].

There is also an SG (Smart Grid) IoT field. The primary grid contains electricity generation, transmission, distribution, and consumption. Electric consumption must be read from a meter by people. On the other hand, SG implements automatic information generation by including in the primary grid smart IoT devices like SG's most crucial device SM (Smart Meter). The SM uses two-way communication so the provider can read data from it, update its software, etc. Consumers can estimate bills from collected information provided using a mobile application or internet site. Providers can limit the maximum electricity consumption or connect and disconnect the load when needed[4].

1.1 Smart Meters

Smart meters measure the consumption of different energies like water, gas, heat, and electricity. The central part of Smart Meter being a Smart Meter is that it can communicate and analyze data. Each Smart Meter can display the measured data on an internal display and also send data to the server for further processing. In each subsection, there are specified the main ones.

1.1.1 Smart Gas Meter

A gas meter is used for measuring the amount of gas that the house or other facility consumes. There are different types of gas meters. As smart gas meters, there are Electronic gas meters and Ultrasonic gas meters. The Electronic gas meter has sensors to measure the pressure and temperature of the gas, and from that, it calculates the gas flow. The Ultrasonic gas meter uses ultrasonic waves to measure the gas flow. Ultrasonic waves are transmitted through the gas and are captured by a sensor. The gas flow is calculated from the time between transmitting and receiving the waves[5].

1.1.2 Smart Water Meter

A water Meter is used for measuring the amount of water a house consumes from the common water source. For smart metering, two types are used electromagnetic and ultrasonic water meters. Electromagnetic water meters are based on Faraday's law of induction, where flowing water acts like a conductor, and meters around the flowing water generate magnetic fields, which generate current. For that, two electrodes detect the amount of generated current. With the change in the flow, there is a change in the intensity and frequencies of current. The water meter calculates the water flow from the measured intensity and frequencies. Ultrasonic water meters use high-frequency sound waves transmitted into the fluid stream. The waves are captured and analyzed by a detector. The water flow is calculated from the time between transmitting and receiving the waves. Each method is accurate and can send data to the central server[6].

1.1.3 Smart Calorimeter

The calorimeter measures the heat energy produced by a chemical reaction or a physical change. For metering, the calorimeter uses sensors that can detect small changes in temperature and pressure. The calorimeter can have advanced software which collects and analyze data in real-time. There are also different types of Calorimeters, like a Flow Calorimeter and an Optical Calorimeter. The flow calorimeter is used to measure the heat output of fluids. It passes the fluid through a cooled or heated part, where the calorimeter measures the temperature change. The Optical Calorimeter is used for measuring the heat output of lasers. The laser is focused on a sample where is measured the temperature increase[7].

1.1.4 Smart Electric Meter

The electric meter is used for measuring house or facility electricity consumption. The electric meters use alternating current sensors that detect current and voltage from the grid. From measured current and voltage, it calculates the watts of consumption. Smart electric meters can support real-time monitoring, automatic reading of values, and remote connection/disconnection of electricity[8].

1.2 Communication Principles

In the following sections, there is shown the difference between different communication principles. The main difference is whether machines or humans are communicating. There is also a historic difference where machines did not need to communicate until the first sign of information-collecting systems or the first automatization.

1.2.1 H2H (Human-to-Human)

Oldest communication principle that we can compare to talking between two people. This communication principle does not help with automatization but is helpful for information exchange. The best examples would be talking on the telephone or sending an email. In the telephone between people, there is no need to guarantee the transmission of transmitting all data because even with slight data loss, people can still understand each other. There are 2G, 3G, and 4G mobile networks for this type of communication, which are not for IoT communication.

1.2.2 M2M (Machine-to-Machine)

Communication between two machines was developed to connect devices with applications. The primary use of M2M communication is automated processing, data generation, transfer, and data exchange between machines. M2M communication was begging for simple IoT structures. Like in the early-stage SCADA systems communicate by a twisted pair because there was none of the wireless systems. After the wireless system became affordable, most devices were designed as purpose-built, handling only particular applications such as baby monitoring or remote-controlled lighting[1].

1.2.3 mMTC (massive Machine-Type Communication)

They are designed for large quantities of simple devices requiring small and infrequent data transfers. mMTC is ready to handle a massive number of devices in large

areas. These devices must have long battery life for up to 10 years. MTC is one of the main communication principles for IoT because almost every type of IoT needs this type of communication. For mMTC communication principles, there is a use of LPWAN communication because of the emphasis on transmission speed, latency, loss rate, coupling loss, and battery lifetime. Examples of mMTC can be sensors of every kind or asset tracking or another[1].

1.3 Legacy Communication Technologies

These technologies were developed for H2H communication, like telephoning or the internet for cell phones. These technologies were modified for use in IoT, which are discussed in the following sections.

1.3.1 The Second Generation

2. Generation of Mobile Networks is mainly used in IoT for low price, low power consumption, and availability. This generation are enough for small and infrequent data transfers. They are primarily used with mMTC for sensing or measuring devices. This generation is still in use through the turn-off of 3G (3. Generation of Mobile Networks). 3G was used for more extensive data transmission, but with the arrival of 4G, there was no need for this type. Moreover, 2G had the upper hand because of its module cost and small battery consumption. 2G supports bandwidth up to 64 kbps, and this speed makes real-time video transmission impossible. There is a risk of 2G mobile networks being shut down, which will probably happen in 2027 or, at the latest, in 2030 in terms of Vodafone[9].

1.3.2 The Fourth Generation

4. Generation of Mobile Networks enables high transmission speed and low latency. It can be used for real-time communication with great demands like cameras or drones. It has excellent coverage but is not the best option for many IoT devices because of the higher module cost and high power consumption. 4G offers an upload speed of up to 50 Mbps. 4G supports a much greater number of users than the previous generation. The maximum bandwidth is 1 Gbps, and as IoT device requirements increase, it can become a modernisation bottleneck. In 3GPP Rel. 12 for 4G, there was the first definition for MTC. After that, in Releases 13-15, comes the first definition for NB-IoT and LTE Cat-M communication technologies with a smaller bandwidth, NB-IoT with bandwidth 200 kHz and LTE Cat-M with 1,4 MHz[9].

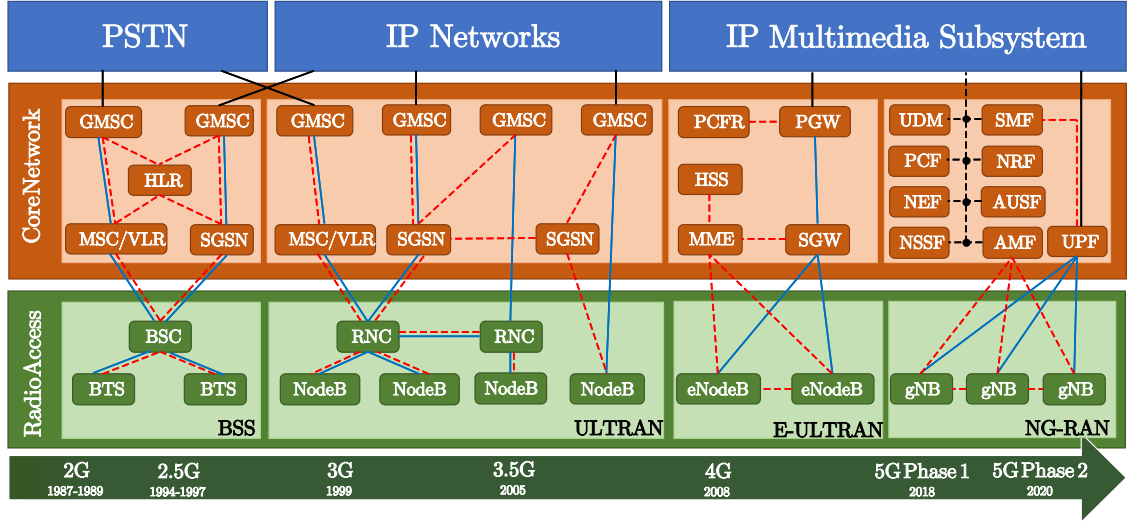


Fig. 1.1: 3GPP Technologies Evolution[10]

1.4 LPWAN (Low-Power Wide Area Networks)

LPWAN is designed to achieve long-range connectivity by using only a tiny amount of energy which is especially good for IoT usage. It uses the principle of one single base station in the use of NB-IoT (Narrow-Band Internet of Things) and LTE Cat-M, and gateways in the use of Sigfox and LoRaWAN (Long-Range Wide Area Networks), which control all connected IoT devices. It supports only low transmission speed. Therefore, it has low power consumption and is suitable for battery usage. In the unlicensed frequency band, there are Sigfox and LoRaWAN. In a licensed frequency band, there are NB-IoT and LTE Cat-M[11].

1.5 Communication Technologies

The following communication technologies all belong to LPWAN. These technologies are designed for use in IoT, mainly for mMTC, and use licensed and unlicensed frequency bands.

1.5.1 Sigfox

Belongs to LPWAN in an unlicensed frequency band. It is designed for IoT with small battery consumption, cheap purchase price, low fees, and long-range connectivity. It can communicate with short messages. It supports 100 Hz and 600 Hz bandwidth. Sigfox claims to support a maximum path loss of 163 dB in the European 868 MHz band. Maximal transmission speed is 100 bps for uplink and 600 bps for downlink. Sigfox uses half duplex mode. Latency stands in the range from 1 to

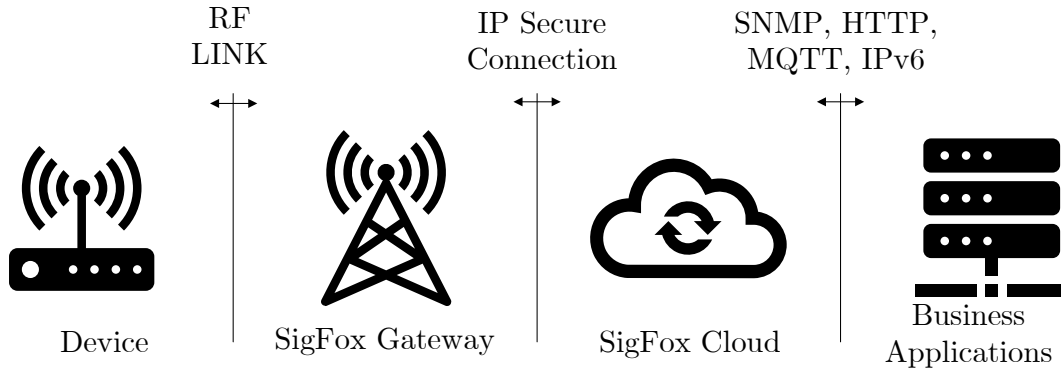


Fig. 1.2: Sigfox Communication Infrastructure[12]

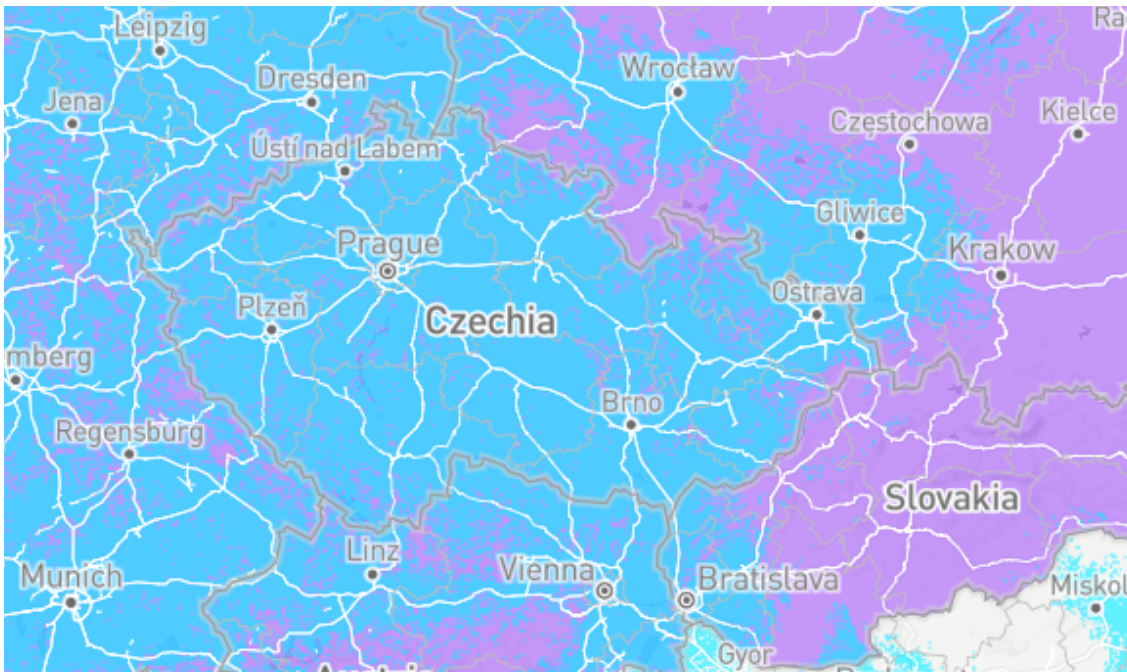


Fig. 1.3: Sigfox Coverage of Czech Republic (Blue - Live Coverage, Ping - Under Roll-Out[13])

30 s. The size of the uplink message is 12 bytes, and for the downlink, 8 bytes. The maximal number of uplink messages per day is limited to 140, for the downlink limit is 4 messages per day. End devices send messages three times in different frequency channels[14, 11].

1.5.2 LoRaWAN

Belongs to LPWAN in an unlicensed frequency band. Network technology is designed to provide long-range connectivity to battery-operated devices. The LoRa claims to support a maximum coupling loss of 155 dB in the European 867 MHz –

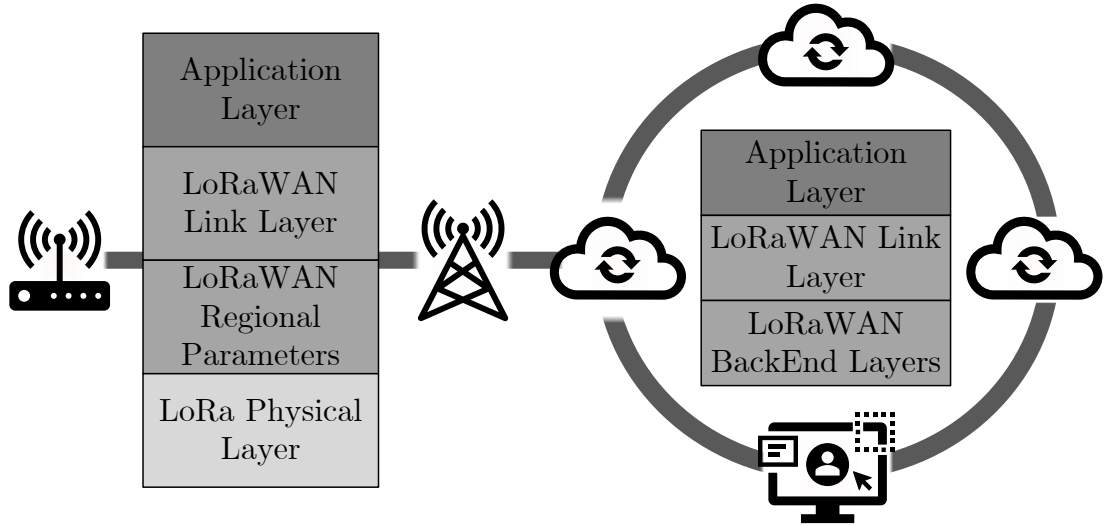


Fig. 1.4: LoRaWAN Communication Infrastructure[15]

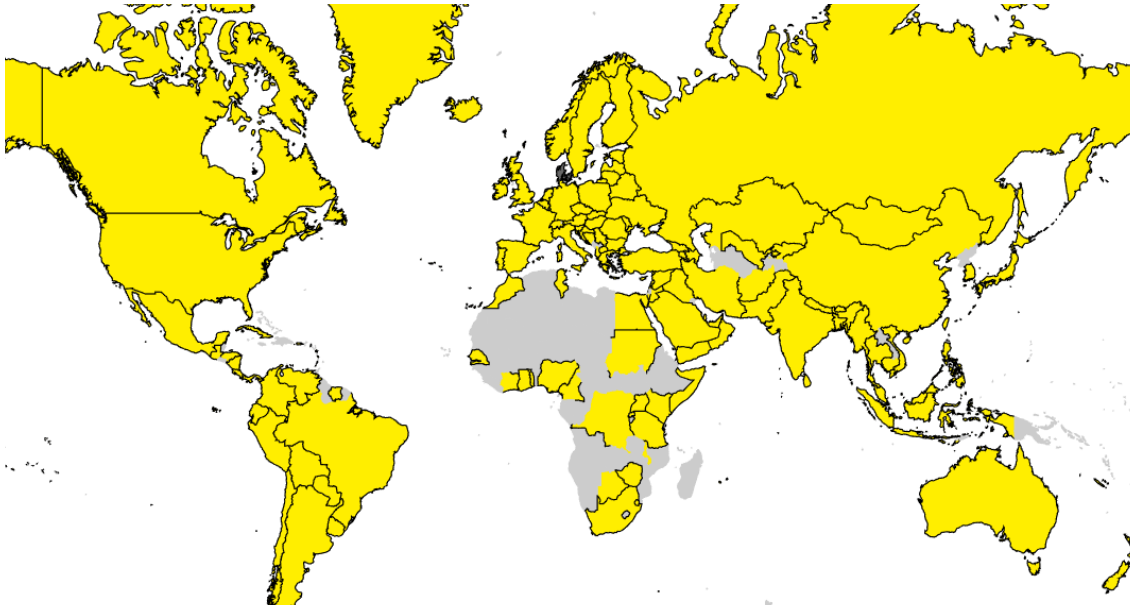


Fig. 1.5: LoRaWAN Coverage of the Whole World[16]

869 MHz band. The channel bandwidth is mainly 125 kHz. Supported data rates lie in the range of 300 bps – 50 kbps. The maximum payload length for each message is 243 bytes. LoRaWAN has multiple communication classes for the different latency in IoT applications based on needed stats[14, 11].

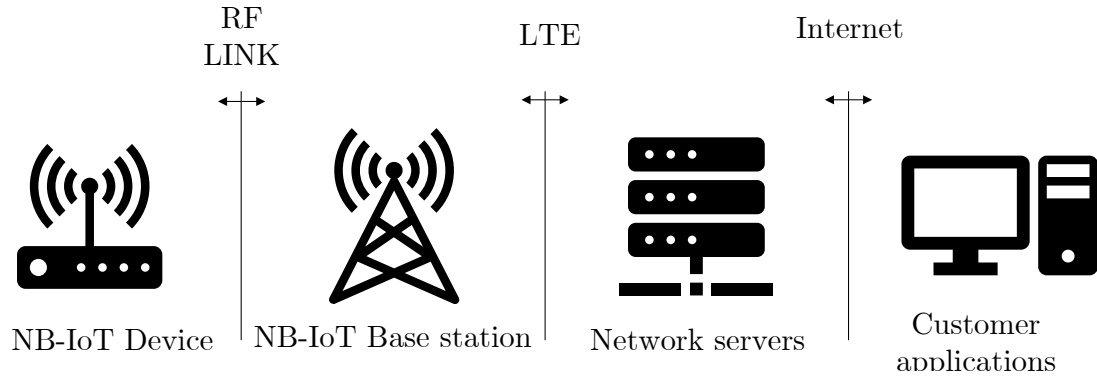


Fig. 1.6: NB-IoT Communication Infrastructure[15]



Fig. 1.7: Vodafone NB-IoT Coverage of Czech Republic[16]

1.5.3 NB-IoT

It belongs to LPWAN in a licensed frequency band, so a SIM card is needed. The NB-IoT communication technology is based on the LTE communication technology. The main feature of NB-IoT is excellent indoor coverage. Must support, like every LPWAN device, a high number of connected devices with low power consumption. One NB-IoT network cell should be capable of connecting large flees of devices, even more than 100000 devices. NB-IoT band depends on the operation mode of NB-IoT communication. It uses 200 kHz bandwidth, 180 kHz for data, and 10 kHz on each side as guard bandwidth[14, 11].

1.5.4 LTE Cat-M

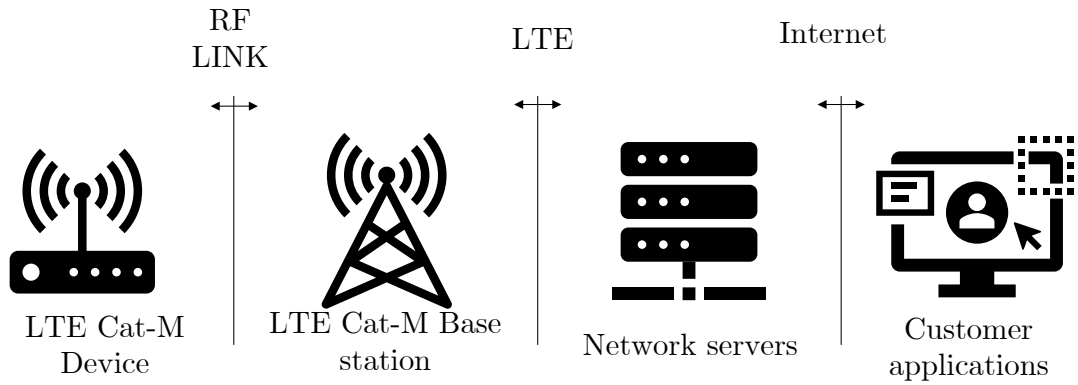


Fig. 1.8: LTE-Cat-M Communication Infrastructure[17]

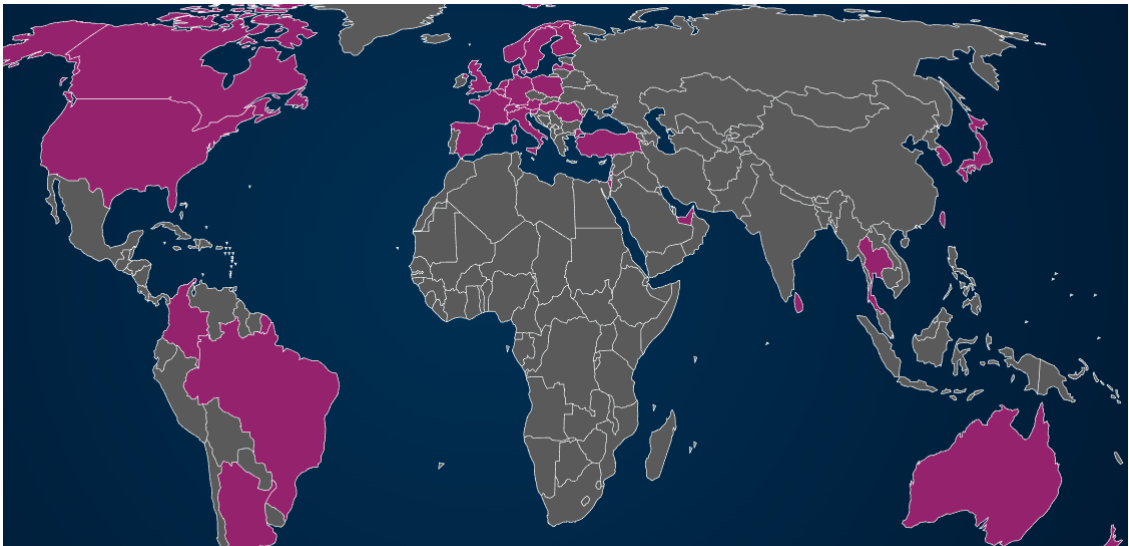


Fig. 1.9: LTE-Cat-M Coverage of the Whole World[18]

It belongs to LPWAN in a licensed frequency band, so a SIM card is needed. The main difference from NB-IoT is low latency. Similar LTE Cat-M has low power consumption, transmits low volumes of data over a long period, and has better penetration of radio waves. Communications protocol allows using full duplex or half duplex communication. Latency stands in the range from 10 to 15 ms. The maximum payload of uplink and downlink messages is the same 1600 bytes[14, 11].

1.6 Comparison of Communication Technologies

Tab. 1.1: Comparison of Communication Technologies[19]

	Sigfox	LoRaWAN	NB-IoT	LTE Cat-M
License	unlicensed	unlicensed	licensed	licensed
Transmission speed for uplink	0,1-0,6 kb/s	0,25-11 kb/s	0,3-62,5 kb/s	HD: 590 kb/s FD: 3 Mb/s
Transmission speed for downlink	0,6 kb/s	0,25-21,9 kb/s	0,5-27,2 kb/s	HD: 800 kb/s FD: 1 Mb/s
Latency	1-30 s	1-10 s	1,6-10 s	10-15 ms
Range	<13 km	<20 km	<15 km	<1 km
Bandwidth	100, 600 Hz	125, 250, 500 kHz	200 kHz	1,4 MHz
Coupling loss	163 dB	155 dB	164 dB	156 dB
Max. EIRP	UL: 14 dBm DL: 27 dBm	14 dBm	23 dBm	23 dBm
Frequency band	868/915 MHz	433/868/915 MHz	700-2100 MHz	700-2600 MHz
Max. payload for uplink	12 B	243 B	1600 B	8188 B
Max. payload for downlink	8 B	243 B	1600 B	8188 B
Battery life	10+ years	10+ years	10+ years	10+ years
Consumption	Tx: 14 mA Rx: 7 mA PSM: < 1 μ A	Tx: 44 mA Rx: 12 mA PSM: < 1 μ A	Tx: 240 mA Rx: 46 mA PSM: 3 μ A	Tx: 360 mA Rx: 70 mA PSM: 8 μ A
Security	AES-128	AES-128	LTE Security	LTE Security
Technology	Proprietary	PHY: Proprietary MAC: Open	Open LTE	Open LTE
Module cost	2\$	6\$	8\$	10\$

2 CIoT (Cellular Internet of Things)

This chapter provides more detailed information about licensed IoT devices like NB-IoT and LTE Cat-M. In the beginning, there is a section about a 3GPP organization that takes care of technical support for all CIoT technologies and releases new ones. Also, there are be information about the CIoT devices' low battery consumption modes[1].

2.1 3GPP (3rd Generation Partnership Project)

The 3GPP unites seven telecommunications standard development organizations representing Europe, the United States, China, Korea, Japan and India. 3GPP specifications cover cellular telecommunications technologies, including radio access, core network and service capabilities, which provide a complete system description for mobile telecommunications. The 3GPP specifications also provide hooks for non-radio access to the core network and interworking with non-3GPP networks. 3GPP has, since its start in 1998, organized its work in release cycles and has, in 2022, reached Release 18. 3GPP follows procedures with sections covering: Description, Participation, Structure, Partners' collective responsibilities, PCG (Project Coordination Group), Technical Specification Groups(Incl. Elections), Work Programme and Technical Coordination, Deliverables(Technical Specifications and Technical Reports) and Reporting. At the end of this process, publishing a 3GPP release. When new features are introduced with a release, new technical specifications for technologies from the previous release are provided. Each release contains new features supporting GSM, UMTS, LTE and NR. Every release must support GSM, UMTS, LTE, and NR to coexist in the same area. The most important thing for 3GPP is forward and backward compatibility. So with new technologies, they still have to support older ones, at least for some time.

Member companies drive the production of specifications and studies at the TSG (Technical Specification Group). The Working Groups meet regularly for their quarterly TSG Plenary meetings, where their work is presented for information, discussion, and approval. PCG is responsible for project management and has the right to change projects. Above PCG are only Organization Partners, which are ARIP (Japan), CCSA (China), ETSI (Europe), ATIS (US), TTA (Korea), TTC (Japan), and TSDSI (India). They hold the ultimate authority to create or terminate TSGs and are responsible for the overall scope of 3GPP.

Most important for LPWAN was the Release 13 massive MTC (mMTC) specification. Work on EC-GSM-IoT, NB-IoT and LTE-M was led by TSG GERAN (GSM

EGPRS RAN) TSG RAN. In this release, a TR (Technical Report) 45.820 Cellular System Support for Ultra-Low Complexity[1, 20] was published.

2.2 Low Battery Consumption Modes

2.2.1 PSM (Power Saving Mode)

When PSM is activated, the device enters a power-saving state, reducing its power consumption to a minimum. In PSM mode, the device stays registered to the network, so it does not need to reconnect to the network first when it leaves the PSM. On the other hand, in PSM mode device is unreachable for communication[21].

2.2.2 eDRX (extended Discontinuous Reception)

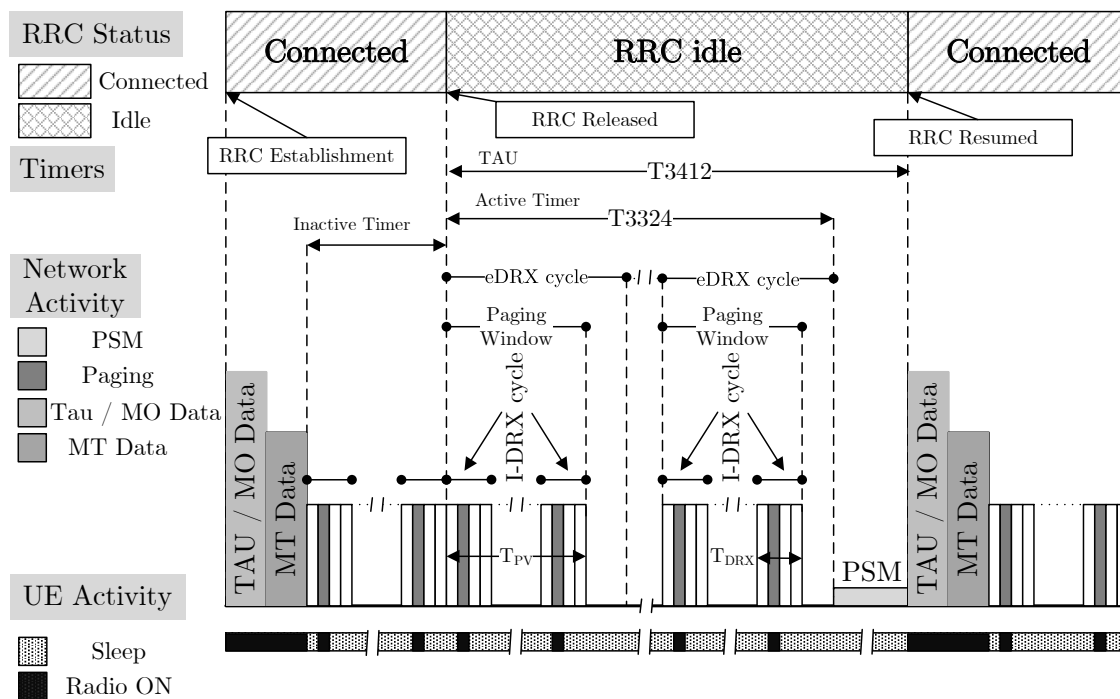


Fig. 2.1: Summary Diagram of CIoT Device in eDRX Mode[22]

The eDRX mode extends the DRX cycles to allow a device to stay in power-saving mode longer. Again PSM mode in eDRX mode, the device is available for mobile-terminal services, reducing the latency for downlink transmissions[21].

2.3 NB-IoT (Narrow-Band Internet of Things)

This section discusses more details about NB-IoT communication technologies used in CIoT, like its mode of operation and channels.

2.3.1 Modes of operation

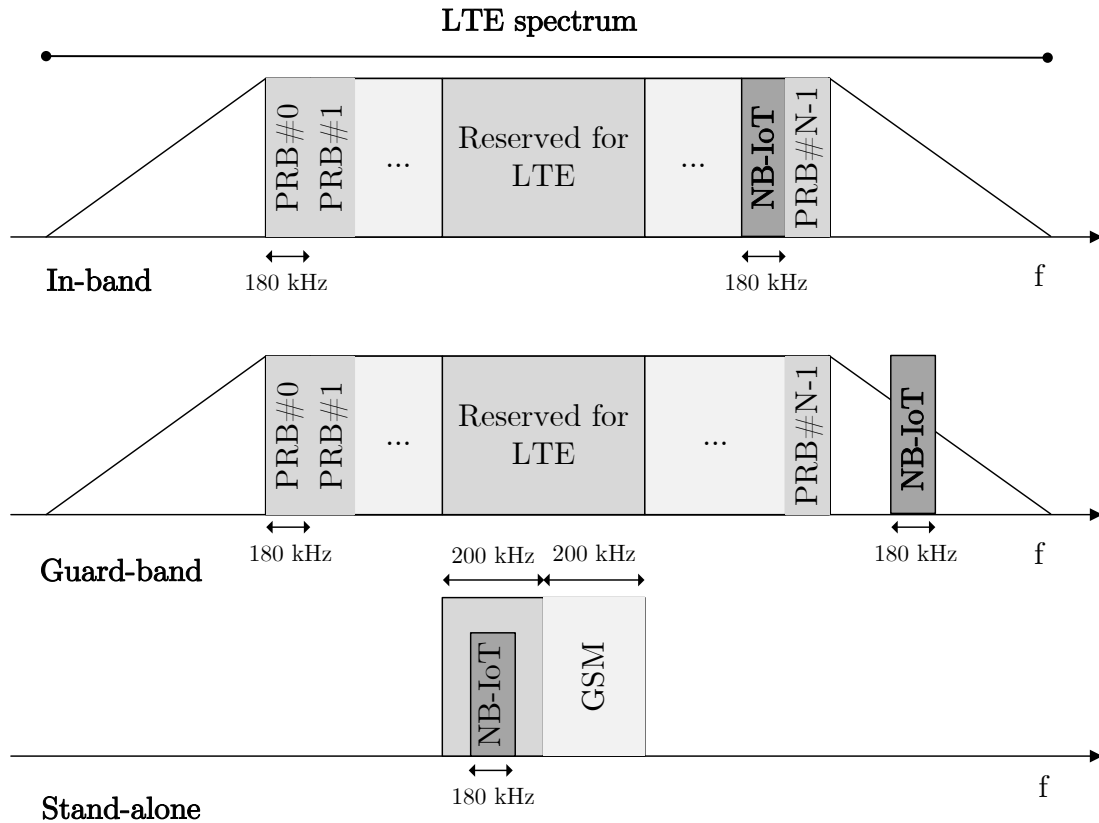


Fig. 2.2: NB-IoT Modes of Operation[23]

In-Band

NB-IoT carrier is placed right into LTE bandwidth. In this case, the NB-IoT carrier is using LTE PRBs (Physical Resource Blocks) that are reserved for NB-IoT[23].

Guard-Band

NB-IoT carrier is placed right into LTE bandwidth. In this case, the NB-IoT carrier is using LTE PRBs that are reserved for NB-IoT[23].

Stand-Alone

Part of the GSM spectrum is replaced with an NB-IoT carrier. The GSM spectrum is divided into 200 kHz GSM carriers. For stand-alone mode, we remake two GSM carriers into one NB-IoT carrier with a 10 kHz guard-band on both sides to prevent interference between these technologies[23].

2.3.2 Communication Channels

NPSS/NSSS (Narrowband Primary Synchronization Signal and Narrowband Secondary Synchronization Signal)

The NPSS/NSSS is used to synchronize devices with the NB-IoT cell. NPSS is transmitted every 10 ms and NSSS every 20 ms. By synchronization with NPSS and NSSS, the device can detect the cell identity number and the framing information[1].

NPBCH (Narrowband Physical Broadcast Channel)

NPBCH is responsible for transmitting the Narrowband MIB-NB (Master Information Block), which provides information for the device to work in the NB-IoT network. Transmission of NPBCH is repeated eight times, where MIB-NB is transmitted without any content change for 640 ms using QPSK modulation to ensure that information is received in extreme coverage conditions. MIB-NP is a 50-bit block that contains 16-bit CRC[1].

NPDCCH (Narrowband Physical Downlink Control Channel)

NPDCCH is used to carry DCI (Downlink Control Information). There can be three types of DCI information. DCI format N0: Uplink Grant information (23 bits). DCI format N1: Downlink Scheduling information (23 bits). DCI format N2: Indicator of paging, SI Update, and scheduling or notification of changes in multicast control channel (15 bits). Based on DCI 16 bits CRC is generated[1].

NPDSCH (Narrowband Physical Downlink Shared Channel)

NPDSCH is scheduled after NPDCCH for at least 4 ms to give time to end-devices to decode NPDCCH. The primary use of NPDSCH is to transmit unicast data. The high-layer data is segmented into one or more TBs, and NPDSCH transmits one TB at a time. NPDSCH is also used for broadcast information as SI messages. NPDSCH is using QPSK to support a bigger size of TB up to 680 bits[1].

NPRACH (Narrowband Physical Random Access Channel)

NPRACH is used by the device to initiate a connection and by the base station to estimate the ToA (Time of Arrival). ToA of NPRACH represents the delay between the base station and the device. ToA helps the base station to determine the TA (Timing Advance) for aligning the received signal for each device. Up to three NPRACH configurations can be used in NB-IoT cells to support devices in different classes. Configurations are separated by the use of different temporal coverage classes[1].

NPUSCH (Narrowband Physical Uplink Shared Channel)

NPUSCH is used to carry uplink user data or control information from higher layers and to send HARQ Ack/Nack for NPDSCH. NPUSCH has two formats: Format 1 is used to send uplink data. Format 2 is used in signalling HARQ Ack for downlink channel NPDSCH. NPUSCH supports large transport blocks, time-domain re-pletion, single-tone/multi-tone transmission, and low peak-to-average-power ratio modulation schemes for single-tone[1].

2.4 LTE Cat-M

LTE-M is an extension of the LTE network. LTE-M is improving features of LTE for better support of MTC.

2.4.1 Transmission Schemes

Downlink and uplink data transmission are the same as in LTE. Therefore, OFDM (Orthogonal Frequency-Division Multiplexing) is used for downlink and SC-FDMA (Single-Carrier Frequency-Division Multiple-Access) for uplink. LTE-M supports FDD (Frequency-Division Duplex) and TDD (Time-Division Duplex) for duplex mode.

In FDD, there are two different frequencies for downlink and uplink. The FDD can be used in devices for FD-FDD (Full-Duplex FDD) operation and HD-FDD (Half-Duplex FDD). In the HD-FDD mode, the devices switch back and forth between receiving and transmitting. HD-FDD has two operation types. First, type A switches back and forth between reception and transmission fast but has to have two separate oscillators for uplink and downlink frequency generation. On the other hand, Type B uses only one oscillator to generate frequency for uplink and downlink. There is an added guard subframe at every switch from uplink to downlink and vice versa, so the device has time to retune its frequency.

TDD uses the same frequency for uplink and downlink transmission. Communication is divided into subframes: a guard period is called a special subframe. The symbols before the special subframe are used for downlink transmission, and those after the special subframe are used for uplink transmission.

LTE-M devices support FD-FDD, HD-FDD operation type B, TDD, or any combination. This means that LTE-M can use full-duplex and half-duplex for a trade-off between device complexity and performance[1].

2.4.2 Communication Channels

PSS and SSS (Primary Synchronization Signal and Secondary Synchronization Signal)

PSS and SSS provide the cell's carrier frequency, frame timing, CP length, duplex mode, and PCID (Physical Cell Identity). PSS and SSS are transmitted periodically so the device can accumulate the received signal for accuracy even in challenging coverage conditions. LTE supports 504 PCIDs divided into 168 groups with three identities. Each PSS sequence contains 168 SSS sequences indicative of the PCID group. In the FDD, PSS is mapped to the last OFDM symbol in slots #0 and #10, and SSS is mapped to the symbol before PSS. In the TDD, PSS is mapped to the third OFDM symbol in subframes #1 and #6, and SSS has mapped three symbols before PSS[1].

RSS (Resynchronization Signal)

The RSS was introduced in 3GPP Release 15 to enhance energy efficiency. RSS is obtaining time and frequency synchronization when a device needs to reconnect. RSS is transmitted less often than PSS/SSS but with more energy, so the RSS signal is more reliable for cells. With the use of RSS, resynchronization time is 40 ms versus over 1 s with the use of PSS/SSS. The first synchronization cell still needs PSS/SSS, but after that can use RSS[1].

CRS (The Cell-Specific Reference Signal)

The CRS is being used for the demodulation of PBCH and PDSCH. The CRS can be transmitted from up to four logical antennas. CRS is mapped to REs in every PRB and every subframe unless CRS muting is used. Muting was introduced in Release 15, and the base station uses it for turning off the CRS signal when the Cat-M device does not need them for demodulation or measurements[1].

DMRS (Demodulation Reference Signal)

The DMRS is used for the demodulation of PDSCH or MPDCCH and is configured for every device. DMRS uses the same logical antenna port as PDSCH or MPDCCH. DMRS can use up to 4 logical antenna ports. The device can distinguish DMRS for the four different antenna ports[1].

PRS (Positioning Reference signal)

The PRS is a broadcast signal for the OTDOA (Observed Time Difference of Arrival). The positioning of a receiver device is calculated based on differences in time arrival between PRS signals. The PRS is transmitted through LPP(LTE Position Protocol). PRS subframe is a pseudo-random sequence that is cell-dependent. LTE-M devices have limited receive bandwidth, so they benefit from a PRS that is mapped over a longer duration in time rather than a wide bandwidth. This is why Release 14 has been introduced, where it is possible to configure PRS. Also, introduce multiple PRS configurations, a first with 20 MHz bandwidth but a short time for LTE devices, a second with 5 MHz bandwidth and longer time for Cat-M2 devices, and a third with 1,4 MHz bandwidth and even longer duration for Cat-M1 devices[1].

PBCH (Physical Broadcast Channel)

The PBCH is used to deliver MIB to the device. MIB gives information to the device on how to operate in the network. PBCH is using repetitions for improved coverage. The network can choose whether to enable PBCH repetitions in a cell. Enabling repetitions is used only in cells with the support of in-depth coverage. PBCH has 40 ms TTI, and 24 bits TBS (Transport Block Size). A 16-bit CRC (Cyclic Redundancy Check) is also attached to TBS. All 40 bits are encoded using the LTE TBCC (Tail-Biting Convolutional Code). The encoded bits are mixed with a cell-specific sequence and segmented into four segments distributed to four consecutive frames. A core part of the PBCH is transmitted as four OFDM symbols in subframe #0 in every frame. When using PBCH repetitions core part is transmitted in FDD by subframes #0 and #9 and in TDD by subframes #0 and #5[1].

MWUS (MTC Wake-Up Signal)

The MWUS was introduced in 3GPP Release 15 to improve battery life. LTE-M devices are primarily in idle mode, periodically monitoring paging occasions. Providing only a 1-bit wake-up indication, shorter MWUS can be transmitted before a device needs to wake up to look for pages to indicate whether the device needs

paging occasions or can enter the sleep mode. MWUS is mapped to two PRB pairs within a six-PRB narrowband. The MWUS sequence is designed the same as for the corresponding NB-IoT signal NWUS, except MWUS is mapped to two PRD pairs instead of one in NWUS[1].

MPDCCH (MTC Physical Downlink Control Channel)

The MPDCCH is used to carry DCI (Downlink Control Information). LTE-M device monitors MPDCCH for uplink power control command, uplink grant information, downlink scheduling information, an indicator of paging or system information update, order to initiate a random-access procedure, notification of changes in a multicast control channel, explicit positive HARQ-ACK feedback. REs in one PRB pair is divided into 16 EREGs (Enhanced Resource Element Groups). MPDCCH can use 2,4 or 6 PRB pairs, which transmission can be either localized or distributed. Localized transmission is more suitable for beam forming, meaning that each ECCE (Enhanced Control Channel Element) comprises EREGs from the PRB pair. However, the distributed transmission provides frequency diversity, meaning that each ECCE comprises EREGs from different PRB pairs. For suitable coverage, multiple ECCEs can be aggregated in an MPDCCH[1].

PDSCH (Physical Downlink Shared Channel)

The PDSCH is used to transmit unicast data. PDSCH transmits packets from higher layers segmented into one or more TB (Transport Blocks) but can only transmit one TB at a time. PDSCH can also broadcast system information, paging messages, and random access-related messages. PDSCH has two modes, A and B. A 24-bit CRC is attached to the TB. In mode A in Release 13, PDSCH is modulated with QPSK or 16QAM. In mode B in Release 13, PDSCH is modulated with QPSK. Coverage improvement can be provided by repeating the subframes. LTE-M supports the following PDSCH transmission modes like single-antenna transmission (supported in both A and B mode), transmit diversity (supported in both A and B mode), closed-loop codebook-based precoding (supported only in mode A), and non-codebook-based precoding (supported only in mode B). Release 14 introduces the possibility of restricting the PDSCH modulation scheme to QPSK in connection mode and the possibility of using a larger maximum channel bandwidth than 6 PRBs. The device with 5MHz maximum PDSCH channel bandwidth supports a maximum downlink TBS of 4008 bits, and a device with a 20 MHz maximum PDSCH channel bandwidth supports a maximum downlink TBS of 27376 bits. Release 15 introduces downlink 64QAM support for PDSCH unicast transmission[1].

PRACH (Physical Random-Access Channel)

The device uses PRACH to initialize connection and allows base stations to estimate the ToA of uplink transmission. PRACH is cell-specific in LTE for possible configuration of mapping the signal. LTE-M introduces PRACH CE (Coverage Enhancement) through up to 128 times repetition of the PRACH structure. Cells can support two modes of CE Mode A, which supports up to 2 PRACH CE levels. Furthermore, mode B supports up to 4 PRACH CE levels. PRACH CE levels can be separated by several options: Frequency domain (different levels have different frequencies), Time domain (different levels have different starting subframe periodicities), and Sequence domain (different levels have different preamble sequence groups)[1].

DMRS (Demodulation Reference Signal)

The device transmits the RS (Reference Signal) to the base station to estimate the uplink propagation channel, perform uplink quality measurements, and issue timing advance commands. The DMRS Signal for PUSCH and PUCCH is transmitted in the SC-FDMA. The bandwidth of the DMRS is 1 PRB for PUCCH and is variable for PUSCH[1].

SRS (Sounding Reference Signal)

The network can use the SRS for transmission sounding of the radio channel. PUSCH and PUCCH are shortened when SRS is used to make room for SRS transmission. In CE mode A, the SRS is supported by both periodic and aperiodic transmission. In CE mode B the SRS transmission is not supported[1].

PUSCH (Physical Uplink Shared Channel)

The PUSCH is used to transmit unicast data. PUSCH transmits packets from higher layers that are segmented into one or more TB but can only transmit one TB at a time. A 24-bit CRC is attached to the TB. In mode A in Release 13, PUSCH is modulated with APSK or 16QAM. In mode B in Release 13, PUSCH is modulated with QPSK. Coverage improvement can be provided by repeating the subframes. Release 14 introduces the possibilities for a new range of PUSCH repetition factors to restrict the PUSCH modulation scheme to QPSK. Release 14 also supports larger uplink TBS for Cat-M1 in mode A, the device transmits antenna selection in CE mode A, and it is possible to use larger maximum channel bandwidth than 6 PRBs for PUSCH in CE mode A. Release 15 introduces PUSCH sub-PRB

allocation in mode A and B, it is allowing smaller allocation than 1 PRB increasing the multiplexing capacity[1].

PUCCH (Physical Uplink Control Channel)

The PUCCH is used to carry the following types of UCI (Uplink Control Information): Uplink SR (Scheduling Request), Downlink HARQ feedback (ACK or NACK), and Downlink CSI (Channel State Information). A PUCCH transmission is mapped to two PRB locations with equal distance to the centre frequency of the LTE system bandwidth. The PUCCH is mapped to the SC-FDMA symbols that DMRS does not use. For coverage, enhancement repetition can be provided up to 128 times in Release 14[1].

3 Supported Application and Security Protocols

In this section, there is a description of the application and security protocols supported in CIoT. Application protocols like MQTT, CoAP, LwM2M, and a unique protocol for energy meters DLMS/COSEM. Security protocols like IPsec and TLS.

3.1 MQTT (Message Queuing Telemetry Transport)

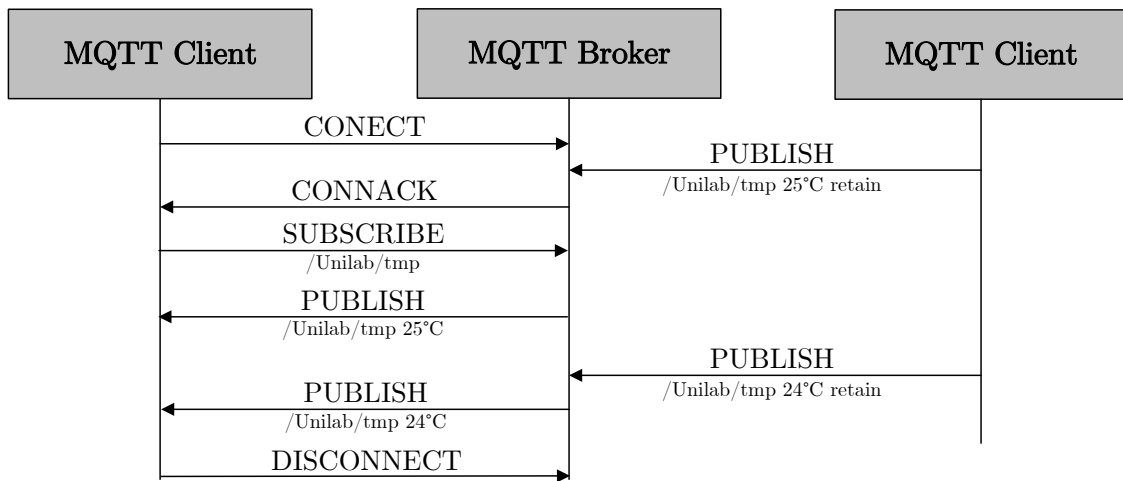


Fig. 3.1: MQTT Communication of Client and Server[24]

MQTT is used as a publish-subscribe messaging paradigm. That means the information sources publish information by sending it to the MQTT broker, and the information sinks can subscribe to certain information by subscribing to the MQTT broker. The information is sorted by topic name. Whenever new information is published on a topic, the broker provides this information to all subscribers. MQTT uses lossless transport protocols like TCP or TLS[1, 24].

3.2 CoAP (Constrained Application Protocol)

CoAP is used for a device with limited power, memory, and processing capabilities. It is a lightweight protocol with a small code size. CoAP is a compacted version of HTTP protocol. CoAP is designed for a network with high packet loss. Devices using CoAP are mostly in battery-saving sleep mode and communicates over constrained-node networks. An IoT device implements a CoAP server that can run on a small

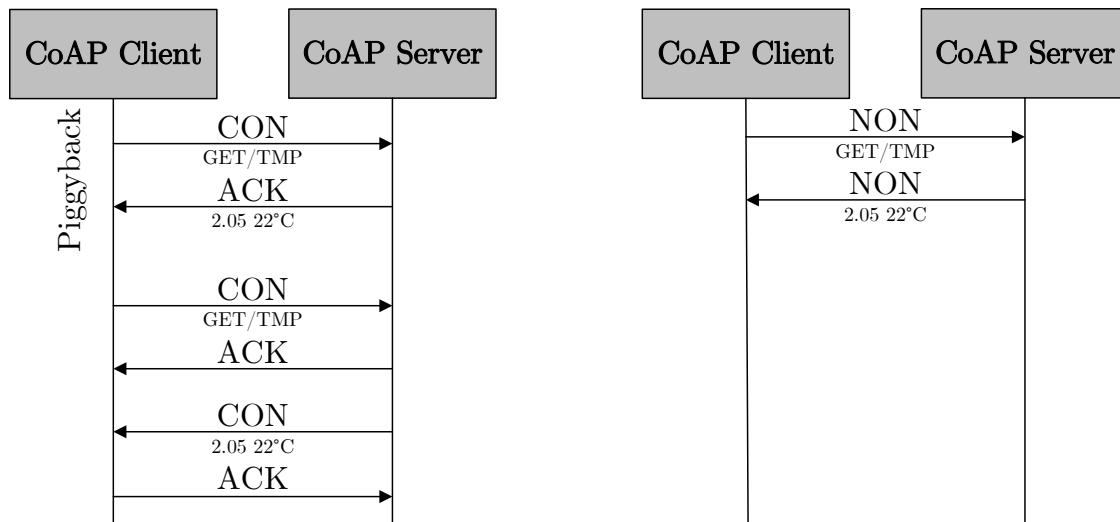


Fig. 3.2: CoAP Communication of Client and Server[24]

computing platform like an 8-bit microcontroller. A CoAP client requests operations on the web resource provider by the sensors. CoAP is using not guarantee successful message delivery via UDP or DTLS protocol. This is why CoAP implements its retransmission mechanism, in which messages can be marked as either confirmable or non-confirmable. Confirmable messages are retransmitted after default timeout until an acknowledgement is received. This method is still lightweight compared to the TCP protocol. Non-confirmable messages are used when communicating with a sensor where the application can tolerate losing some of the readings[1, 24].

3.3 LwM2M (Lightweight Machine-to-Machine)

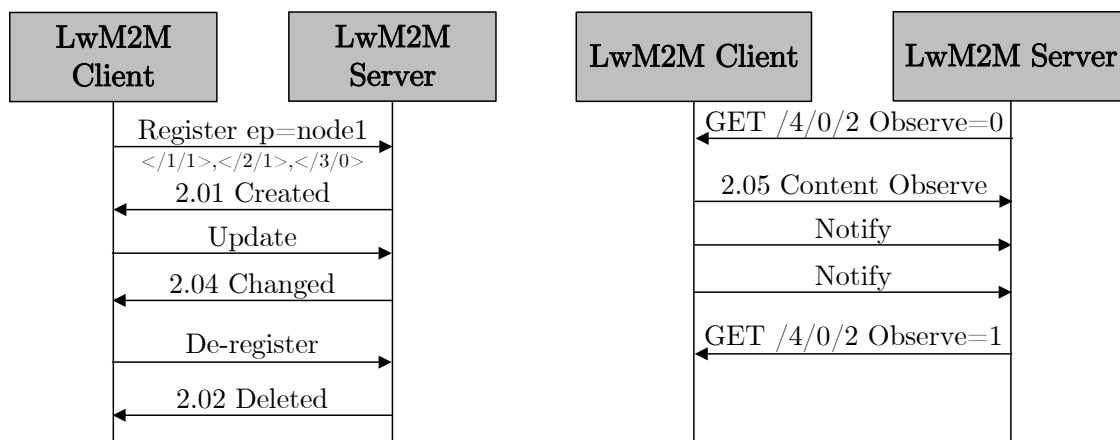


Fig. 3.3: LWM2M Registration and Communication of Client and Server[24]

The LwM2M is a data and management protocol for IoT devices, and it uses CoAP as a transfer protocol. LwM2M defines IoT devices as separate objects representing different things in the device, like sensors or controllers. Every object has resources that describe its properties. Resources can be, for example, data from sensor measurements or device configuration parameters. LwM2M server can communicate with the objects and resources of the device: it can read, write, etc. LwM2M also supports firmware updates of devices, which are transferred through CoAP. LwM2M specifies operations for a device. Using LwM2M in IoT applications is unnecessary, but it gives an application set of functions to simplify the development through the standardized and open framework[1, 24].

3.4 IPSec (Internet Protocol Security)

IPSec secures communication at the IP layer of communication. IPSec devices do not handle encryption and use less processing power and battery. IPSec has two modes transport and tunnel mode. Transport mode secures the transmission of data. On the other hand, a tunnel secures the whole traffic channel between the device and the server. It secures all data from all devices in the network[25].

3.5 TLS (Transport Layer Security)

The TLS secures communication at the application layer. At the beginning of TLS-secured communication, there is a need for a handshake between devices. The handshake mechanism needs back-and-forth messaging, so it increases battery consumption. It uses certificates on the client and server side. The certificates ensure the security of the connection and prevent a man-in-the-middle attack. TLS secures only communication from and to the application. Other devices remain insecure[26].

3.6 DLMS/COSEM (Device Language Message Specification/Companion Specification for Energy Metering)

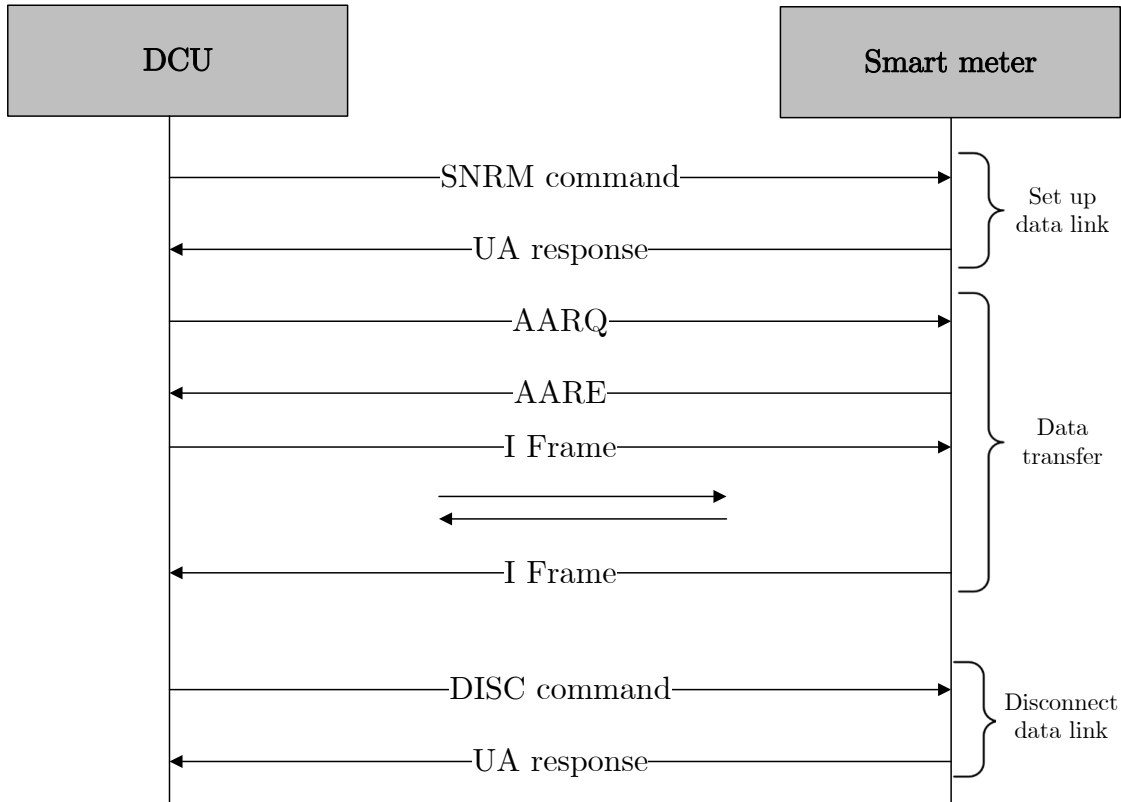


Fig. 3.4: DLMS/COSEM Communication Example[27]

The DLMS/COSEM is a standard energy and water measurement, management, and control protocol. DLMS is a group of open standards developed for data exchange for remote and cable purposes of energy sources, water, gas, and heat. COSEM provides object model definition for any measuring device. DLMS/-COSEM communication starts with setting up a data link between the DCU (Data Concentrator Unit), data transfer, and disconnecting the data link 3.4. These systems are used as client-server paradigm, where the smart meter is the server, and the concentrator is the client[27, 28, 29].

3.6.1 DLMS/COSEM Three-Step Approach

The DLMS/COSEM follows a three-step approach which can be found in the specification.

- The first step is Modelling which covers the interface model of metering equipment and rules for data identification.
- The second step is Messaging, which covers the services for mapping the interface model to APDUs (Application Protocol Data Units) and encoding these APDUs.
- The third is Transporting, which covers the transportation of the messages through the communication channel[30].

3.6.2 DLMS/COSEM Objects

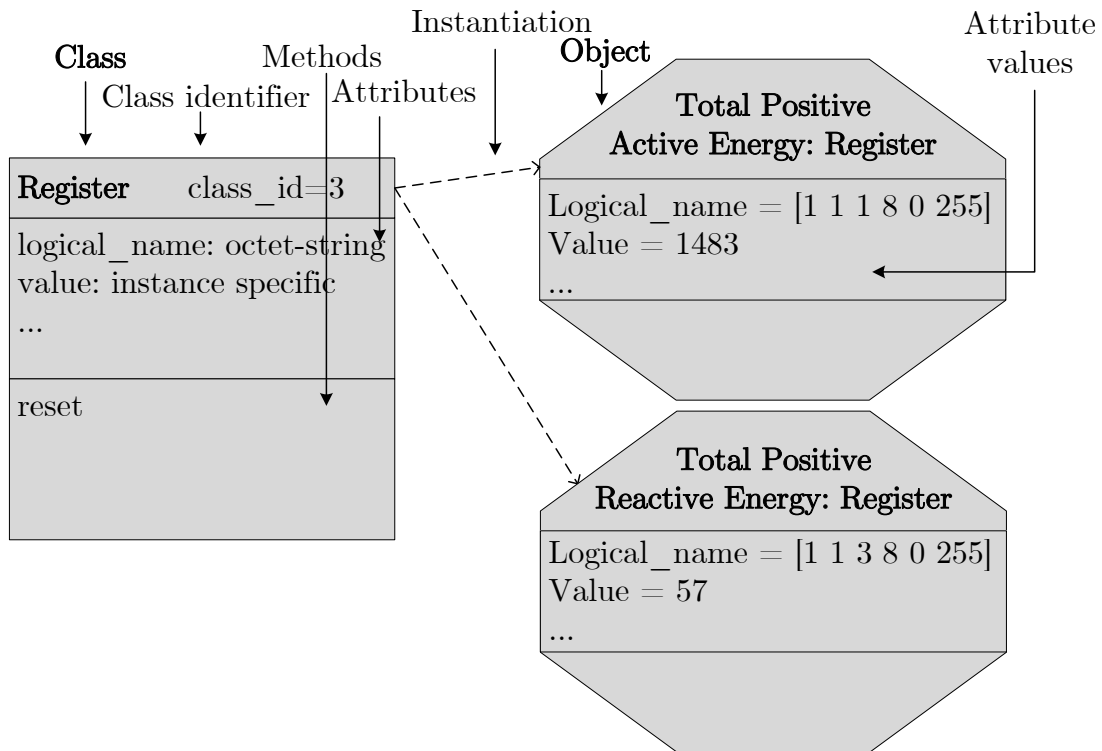


Fig. 3.5: DLMS/COSEM Class and Object Examples[31]

Modelling uses objects, which are collections of attributes and methods. Attributes represent the character of the object. Their value may affect the behaviour of an object. The first attribute of an object is the logicalname which is part of the identification of the object. An object may use several methods to modify the values of the attributes. Objects with common characteristics are generalized as an interface class, identified with a classid[31].

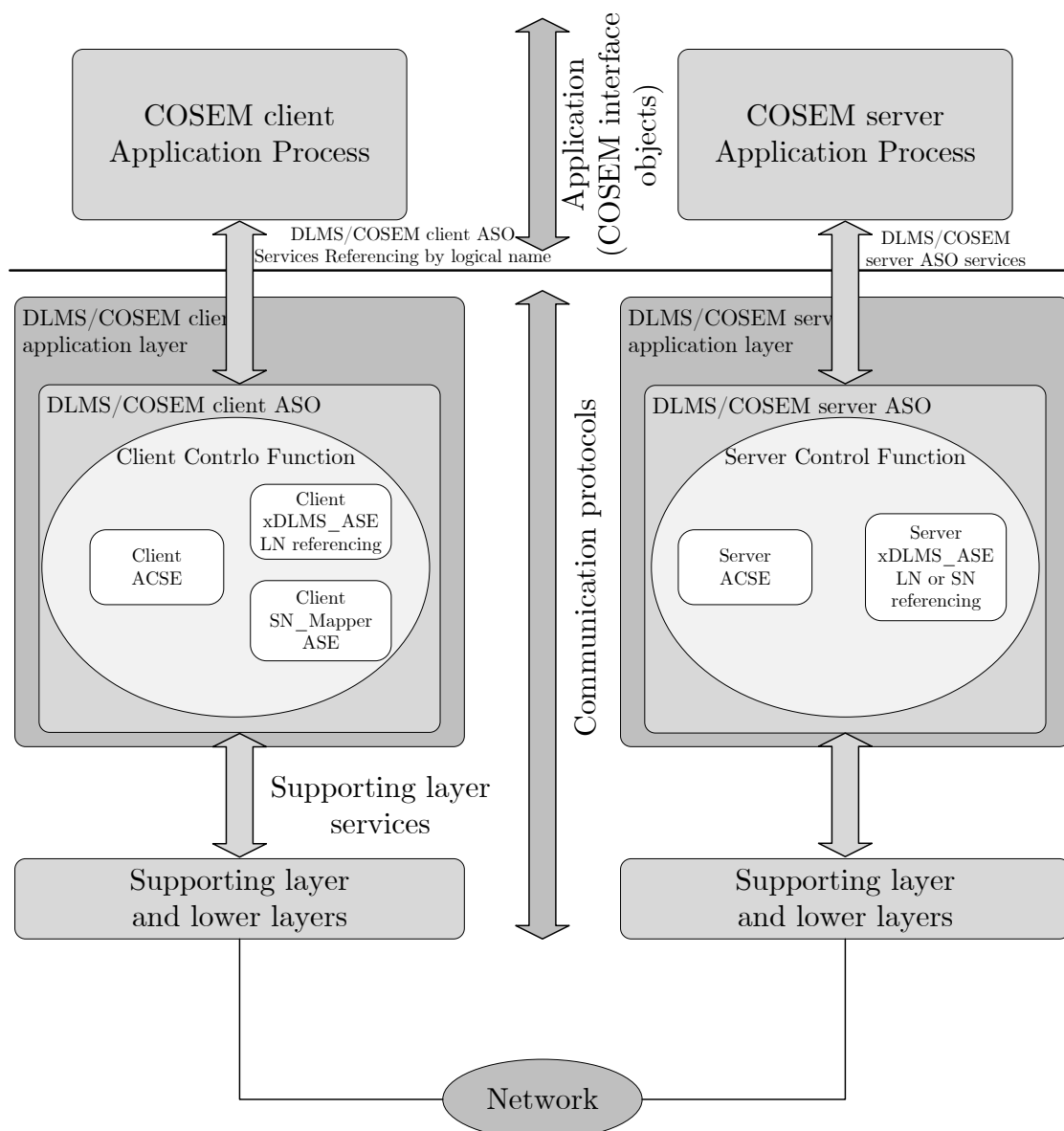


Fig. 3.6: The Structure of the DLMS/COSEM Application Layers[30]

3.6.3 DLMS/COSEM Application Layer

The main component of the application layer is the ASO (Application Service Object). It provides services for users and APs (Application Processes) and uses services provided by the supporting layer. On the client and the server side, it contains three components:

- ACSE (Association Control Service Element);
- cDLMS ASE (extended DLMS Application Service Element);
- CF (Control Function).

Also, there can be a fourth one on the client side, an optional element, the Client

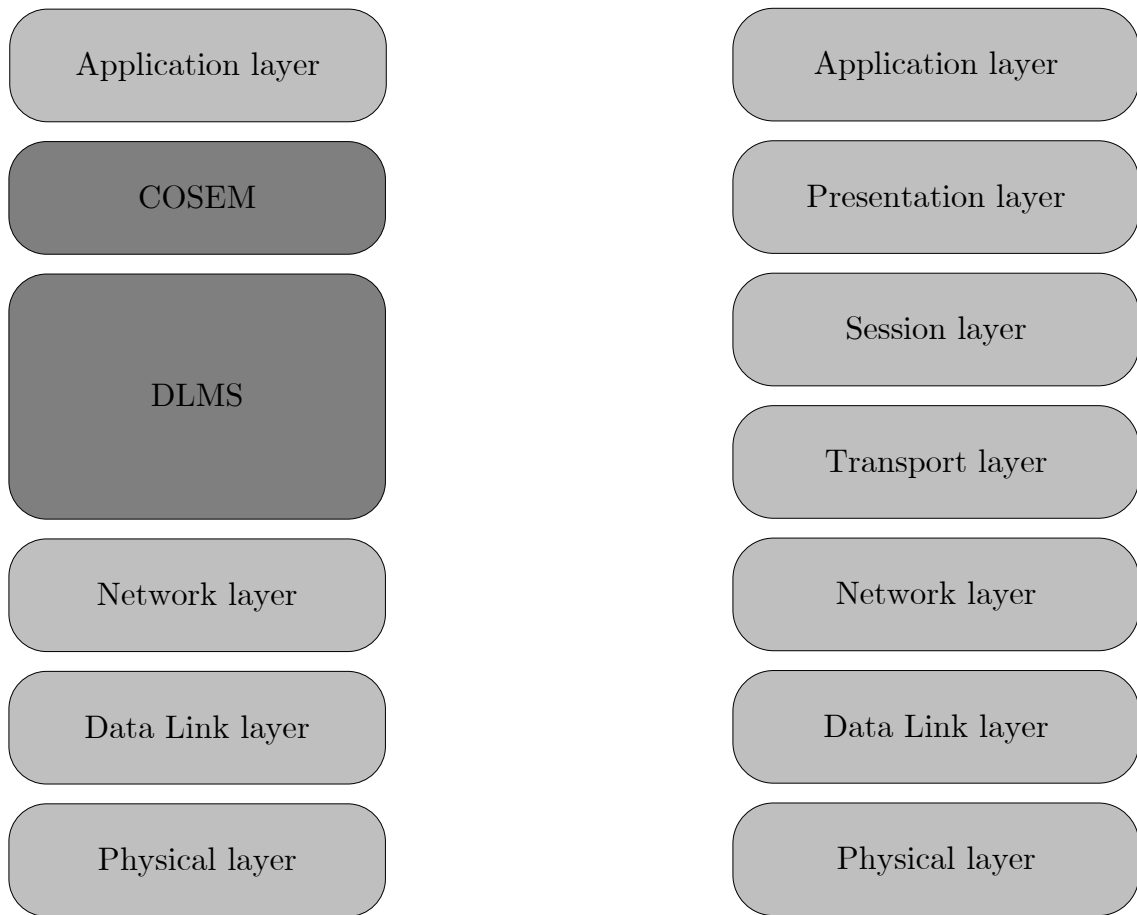


Fig. 3.7: DLMS/COSEM on ISO/OSI Model[30]

SN_Mapper ASE. The ACSE provides services for establishing and releasing AA (Application Associations). xDLMS ASE provides services to transport data between COSEM Aps. CF specifies how the ASO services invoke the appropriate service primitives of the ACSE, the xDLMS ASE, and the services of the supporting layer. SN_Mapper ASE provides a mapping between services using LN and SN referencing. The DLMS/COSEM AL also performs the function of the OSI presentation layer:

- encoding and decoding the ACSE APDUs and the xDLMS APDUs;
- alternatively, generating and using XML documents representing ACSE and xDLMS APDUs;
- applying compression and decompression;
- applying, verifying and eliminating cryptographic protection[30].

3.6.4 DLMS/COSEM Messaging

In confirmed AAs:

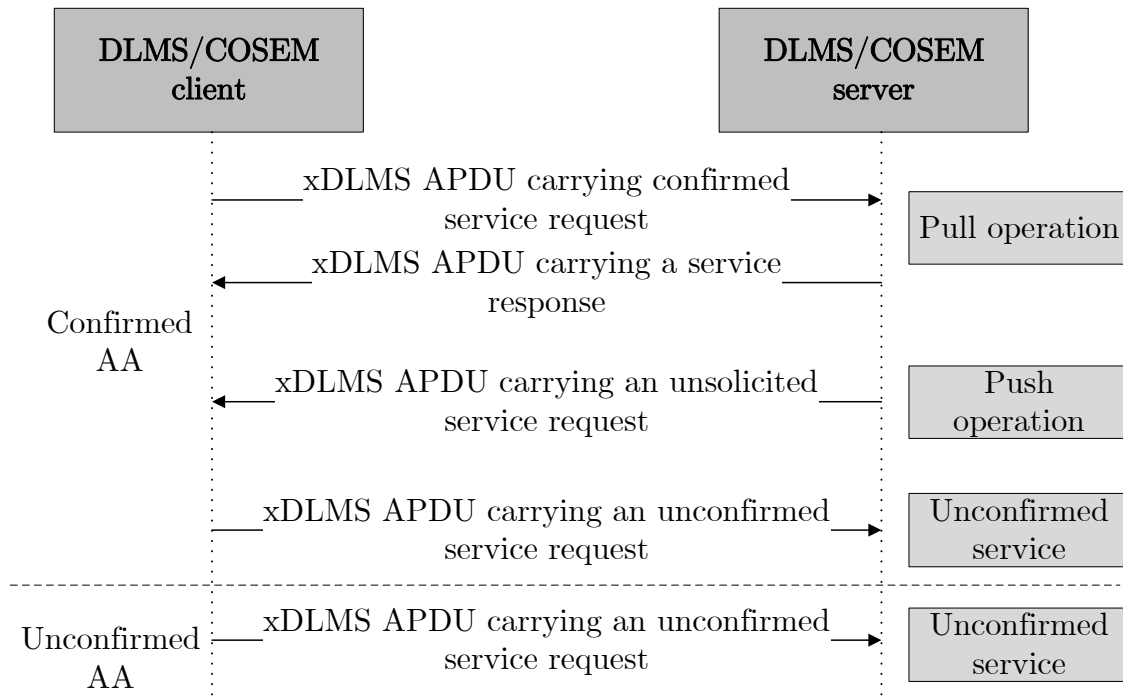


Fig. 3.8: DLMS/COSEM Messaging Patterns[30]

- the client can send confirmed service requests, and the server responds: pull operation;
- the client can send unconfirmed service requests. The server does not respond;
- the server can send unsolicited service requests to the client: push operation.

In unconfirmed AAs, only the client can initiate service requests and only unconfirmed ones. The server cannot respond, and it cannot initiate service requests[30].

3.6.5 DLMS/COSEM Communication Profiles

Communication profiles specify how DLMS/COSEM AL is supported by the lower, communication media-specific protocol layers. Communication profiles comprise many protocol layers, each providing services to its upper layer and using services of its supporting layers. The number of lower layers depends on the communication media used. A single physical device may support more than one communication profile. The client-side AP has to decide which communication profile should be used. Description of the DLMS/COSEM communication profile:

- the COSEM object model modelling the Application Process;
- the DLMS/COSEM application layer;
- the DLMS/COSEM transport layer;
- the convergence layers that bind the MAC layer to the DLMS/COSEM AL;
- the media-specific physical and MAC layers;

- the connection managers[30].

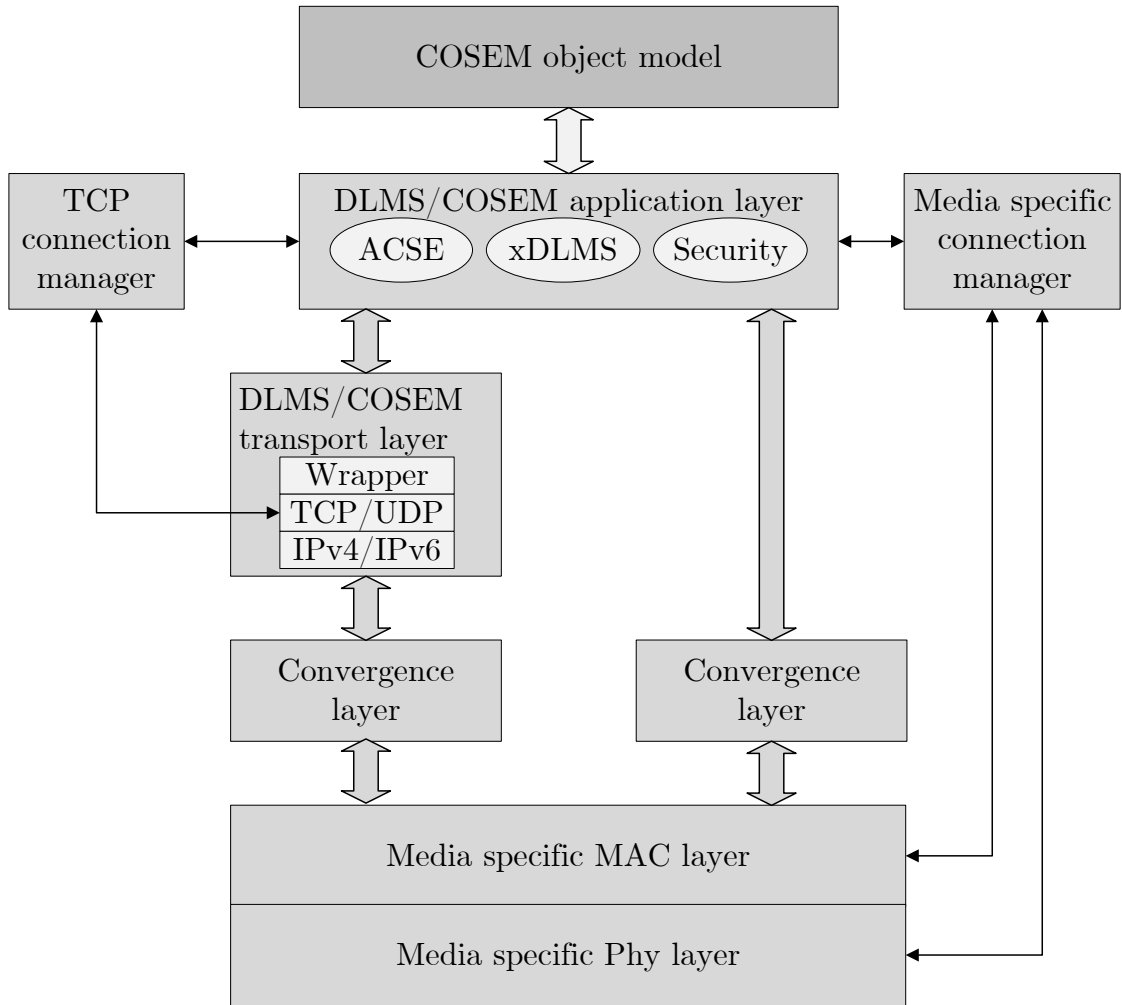


Fig. 3.9: DLMS/COSEM Communication Profile[30]

3.6.6 AAs (Application Associations)

AAs are logical connections between a client and a server. The AAs can be established by client request or can be pre-established. A logical device may support one or more AAs, each with a different client. Confirmed AAs are proposed by the client and accepted by the server when:

- the server knows the user of the client;
- the application context proposed by the client is acceptable for the server;
- the authentication proposed by the client is acceptable for the server and authentication is successful;
- The elements of the xDLMS context can be successfully negotiated between the client and the server.

The client proposes unconfirmed AAs, and the server does not have to accept them. There is no negotiation. It can send a broadcast message from the client to the servers[30].

3.6.7 DLMS/COSEM Security

During the first connection, the client and the server must identify themselves through the AA establishment. The server can also require the identification of the client user. The client and the server may require authentication one from the other[30].

Authentication Mechanisms

The authentication process is used to establish whether the claimant of a specific identity is who he claims to be. Authentication happens during AA establishment. In confirmed AAs establishment, the client or both the client and the server can authenticate themselves. In an unconfirmed AA establishment, only the client can be authenticated. In pre-established AAs, authentication of subjects is not available[30]. The authentication can be specified in three types:

1. No security (Lowest Level Security) authentication
 - The authentication allows the client to retrieve some basic information from the server.
 - Doesn't require any authentication[30].
2. LLS (Low-Level Security) authentication
 - The client authenticates itself by supplying a password known by the server.
 - The client transmits the password to the server using COSEM-OPEN.
 - If the server accepts the password, the client is authenticated, and the AA can be established. If not, the AA is rejected[30].
3. HLS (High-Level Security) authentication
 - The client and server must successfully authenticate themselves to establish an AA. The HLS uses a four-pass process:
 - The client transmits a challenge CtoS and additional information to the server.
 - The server transmits a challenge StoC and additional information to the client.
 - The client processes StoC and additional information. According to the rules of the HLS, and sends the result to the server. If the result is correct, the server accepts the client's authentication.

- The server processes CtoS and additional information According to the rules of the HLS and sends the result to the client. If the result is correct, the client accepts the authentication of the server[30].

4 Practical Part

In this chapter are achieved the first communication through the 5G network. First is some essential information about the used modules and their infrastructure. The second part will be about measured radio properties, and the last part will be about practical use with DLMS/COSEM.

4.1 Devices Used for Communication

More information about devices used for final communication is given in this part.

4.1.1 QUECTEL LPWA BG77 Cat M1/NB2

BG77(4.1) is an ultra-compact LPWAN module that supports LTE Cat-M and NB1-IoT/NB2-IoT bands. It is compatible with the 3GPP Rel-14 specification. Its main advantages are a compact design with ultra-low power consumption, integrated RAM and flash in baseband chipset, hardware-based security features, and support of LGA packages. It has an ARM Cortex A7 processor, which supports ThreadX. This processor can achieve up to 70% reduction of power in PSM mode and up to 85% in eDRX mode. BG77 is mounted on a 5G laboratory unique desk connected to Raspberry Pi Compute module 4. BG77 is operated via a connected Raspberry Pi, whereas typically, BG77 uses AT commands for operating. The maximum data rate in LTE Cat-M1 mode is 588 kbps for downlink and 1119 kbps for uplink. Maximum data rate in NB-IoT mode is 32 kbps for downlink and 70 kbps for uplink, and in NB2-IoT mode, it is 127 kbps for downlink and 158,5 kbps for uplink. Max output power is 23 dBm. Available LTE frequency bands for both LTE Cat-M1 and NB-IoT are B1, 2, 3, 4, 5, 8, 12, 13, 18, 19, 20, 25, 28, and 66. Only NB-IoT uses band 71, especially for LTE Cat-M are bands 26 and 27[32].

4.1.2 RPI CM4 (Raspberry Pi Compute Module 4 Lite)

The RPI CM4 Lite module(4.3) is directly connected to the 5G laboratory unique desk(4.2) and the QUECTEL-BG77 module. For connection, it uses two 100-pin high-density connectors. The primary purpose of RPI CM4 Lite is to operate the BG-77 module. It is controlled using a UART which is connected to a computer via USB. It uses Broadcom BCM2711, a quad-core Cortex-A72 processor. RPI CM4 Lite support 28 GPIO pins, for example, 5 of UART, 5 of I2C, 5 of SPI, etc. DietPi is used as software for RPI CM4 Lite[33].



Fig. 4.1: QUECTEL LPWA BG77 Cat M1/NB2[32]



Fig. 4.2: 5G Laboratory Special Desk

4.2 AT Commands

AT Commands are used in Quectel devices for their programming. All the command lines must start with AT and end with <CR> (Carriage return character). Codes results and responses always start and end with <CR> and <LF> (Line feed character), as follows <CR><LF><response><CR><LF>[34].



Fig. 4.3: Raspberry Pi Compute Module 4 Lite

AT+CPIN

The command is used for sending a password to the MT. The password is necessary for device operation. The password can be (U)SIM PIN, (U)SIM PUK, etc. It also can be used for getting information about the connection of (U)SIM[34].

AT+QCFG

The command is used for extended configuration of settings. The command is used to set the frequency band with `AT+QCFG="band"` command or configure network category with `AT+QCFG="iotopmode"` command, etc[34].

AT+CGDCONT

The command is used for the configuration of APN. It specifies PDP context parameters for a specific context<cid>. The standard layout of this command is `AT+CGCOND=<cid>,<PDP_type>,<APN>`[34].

AT+COPS

The command is used for the selection of the operator. It returns the current operators and their status and allows automatic or manual network selection. It can be used in three modes. The Test command `AT+COPS=?` returns a set of five parameters about the network. The Read command `AT+COPS?` returns the current mode and the currently selected operator. The Write Command `AT+COPS=<mode>`,

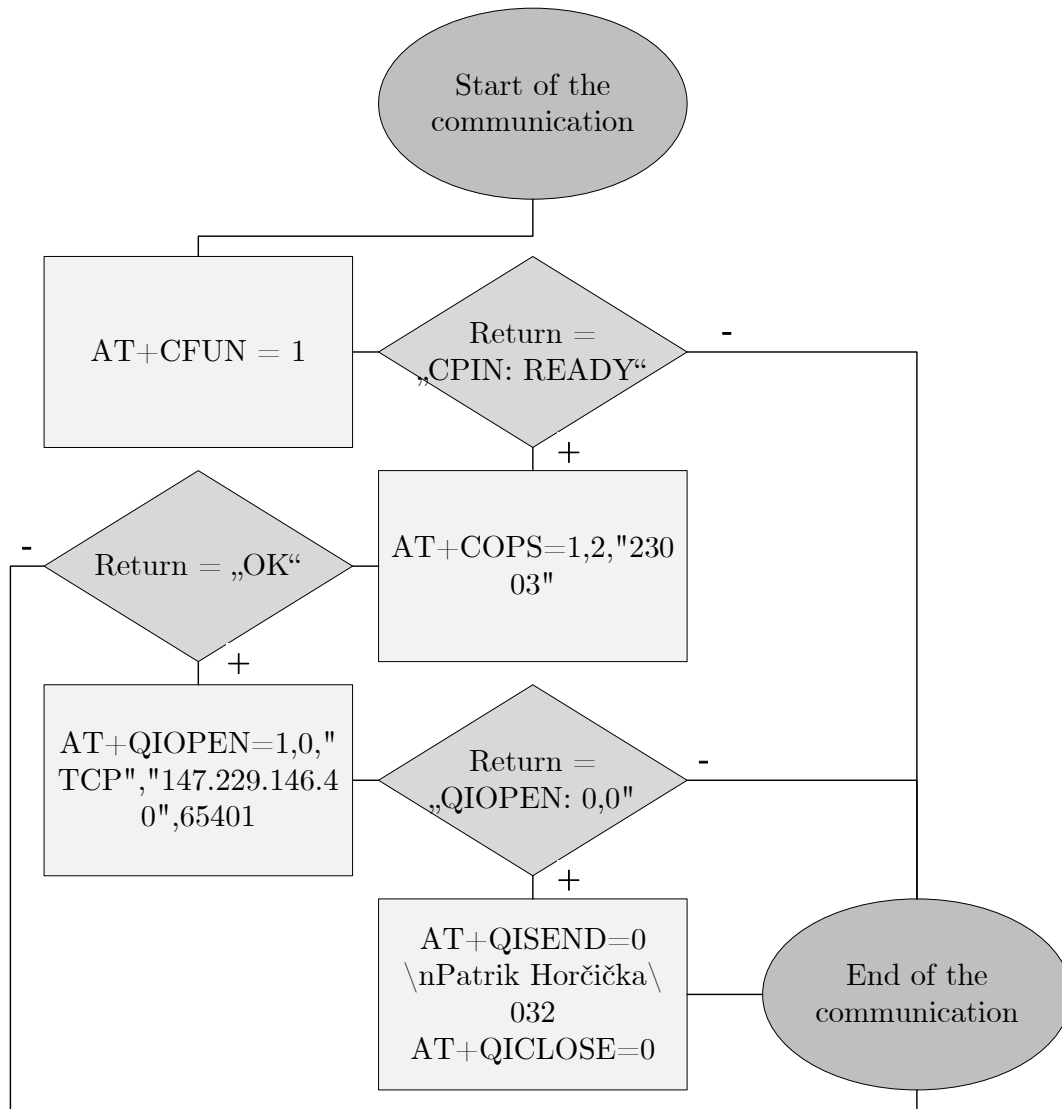


Fig. 4.4: AT-Command Flowchart[34]

<format>, <oper>, <Act> forces an attempt to select and register to the LTE network operator[34].

AT+CEREG

The command is used for getting information about the EPS network registration status. It controls the presentation of an unsolicited result code +CEREG <stat> when <n>=1 and there is a change in the MT's EPS network registration status in E-UTRAN, or unsolicited result code +CEREG: <stat>, <tac>, <ci>, <AcT> when <n>=2 and there is a change of the network cell in E-UTRAN[34].

AT+QCSQ

The command is used for reporting the signal strength of the current service network in the form of AT+QCSQ=?. The command can retrieve the signal strength even if it is not registered. If the MT is not using any service network or the service mode is, the uncertain return value of the query is "NOSERVICE"[34].

AT+QIOPEN

The command is used for opening a socket service. The type of the service can be specified by <service_type> part of the command, and data access mode can be specified by <access_mode> part of the command. The response came as +QIOPEN: <connectID>,<result> if the opening of the socket service was successful[34].

AT+QICLOSE

The command is used for closing specified socked services. The response takes max. 10s and return OK or ERROR messages. Other commands can be sent only after the response is returned[34].

AT+QISEND

The command is used for sending data through the specified network. It can use buffer access mode or direct push mode. An acknowledgement message occurs when the module receives data successfully that is SEND OK. Otherwise, SEND FAIL or ERROR is returned[34].

4.3 Radio Properties for Final Communication

This section provides information about signal properties, speed properties, and delay properties measured in two locations across the Czech Republic. Somewhere one location has excellent radio properties, and the other has bad.

Tab. 4.1: NB-IoT Boundary Condition

ECL 0	SINR >7 && RSRP >-110
ECL 1	SINR <7 && RSRP >-133
ECL 2	SINR <-3 && RSRP <-133

4.3.1 Signal Properties

For measuring signal properties integrated daemon is inside the module. The daemon sends AT commands every 10 s, which return these pieces of information. In both measurements is visible CellID, which tells us the exact number of used BTS (Base Transceiver Station) and other information like BAND, BER, L4, etc.

Brno

This measurement was taken in Brno on Listovy Koleje. As can be seen in 4.2, the signal is solid and stable at this location. It is so because BTS is close to the measuring device, which is confirmed by low TA value and has no interference. From $TA = 1$, it can be calculated that the range between the device and the BTS is 78 m. In the table, we can also see the power of a signal as RSSI that has -63 dBm which is a high value for communication, and Signal-to-Interference-plus-Noise Ratio as SINR that has 11 dB which is also a high enough value for regular communication and also it meets the requirements of ECL 0 which can be seen in 4.1.

Tab. 4.2: Signal Properties in Brno

TA	1
BAND	LTE BAND 20
MCC	230 (DEC)
MNC	3 (DEC)
freq_ch	6447
TAC	48024 (DEC), 0xBB98 (HEX)
RSSI [dBm]	-63
Q_RSRQ [dB]	-12
Q_SINR [dB]	11
Q_RSRP [dBm]	-77
Q_RSSI [dBm]	-64
BER [%]	99
ECL	0
CellID	791839 (DEC), 0xC151F (HEX_STR)
L4	TCP

Heřmanice

This measurement was taken in Heřmanice. As can be seen in 4.3, the signal is weak and unstable at this location. The value of TA is 50, which is an enormous value

and means that BTS is far away from the measuring device, which means a more significant delay in communication. From $TA = 50$, it can be calculated that the range between the device and the BTS is 3906 m. Furthermore, we can see RSSI with -99 dBm, a smaller value than RSSI in Brno, meaning more minor received power. Moreover, SINR has -3 dB, a minimal value for SINR and can bring many interferences. Therefore, it meets the requirements of ECL 1, which can be seen in 4.1.

Tab. 4.3: Signal Properties in Heřmanice

TA	50
BAND	LTE BAND 20
MCC	230 (DEC)
MNC	3 (DEC)
freq_ch	6447
TAC	48093 (DEC), 0xBBDD (HEX)
RSSI [dBm]	-99
Q_RSRQ [dB]	-15
Q_SINR [dB]	-3
Q_RSRP [dBm]	-113
Q_RSSI [dBm]	-99
BER [%]	99
ECL	1
CellID	337951 (DEC), 0x5281F (HEX_STR)
L4	TCP

4.3.2 Measured Speeds and Sizes

The iperf3 software was used to measure the speed of the connection between the online server and the measuring device. First is necessary to start Iperf on the server to listen to the same port, next, Iper on the client with a specified address, and the port can start sending data. Iperf works similarly to PING, it sends packages to specified addresses, but unlike ping, it calculates not only the time between send and receive but also can calculate the bitrate, the size of transmitted data, and the size of the bandwidth for data transfer.

Brno

We can see that transported data in charts 4.5 and 4.6 is linearly growing with sending time. On the contrary, the bitrates in charts 4.7 and 4.8 stay almost the

same. It is due to a more stable connection.

The diagrams 4.5 and 4.6 show the measurement for 5, 10, 15, 30 s of the transmission duration. We can see that the sender sends data in the range between 50 kB and 340 kB. On the other hand, the receiver sends data in the range between 25 kB and 200 kB.

In the figure 4.7 and 4.8, there is also a measurement for 5, 10, 15, 30 s of sending time. We can see that the sender's bitrate is in the range between 45 kb/s and 65 kb/s. Then again, the receiver's bitrate is in the range between 36 kb/s and 65 kb/s.

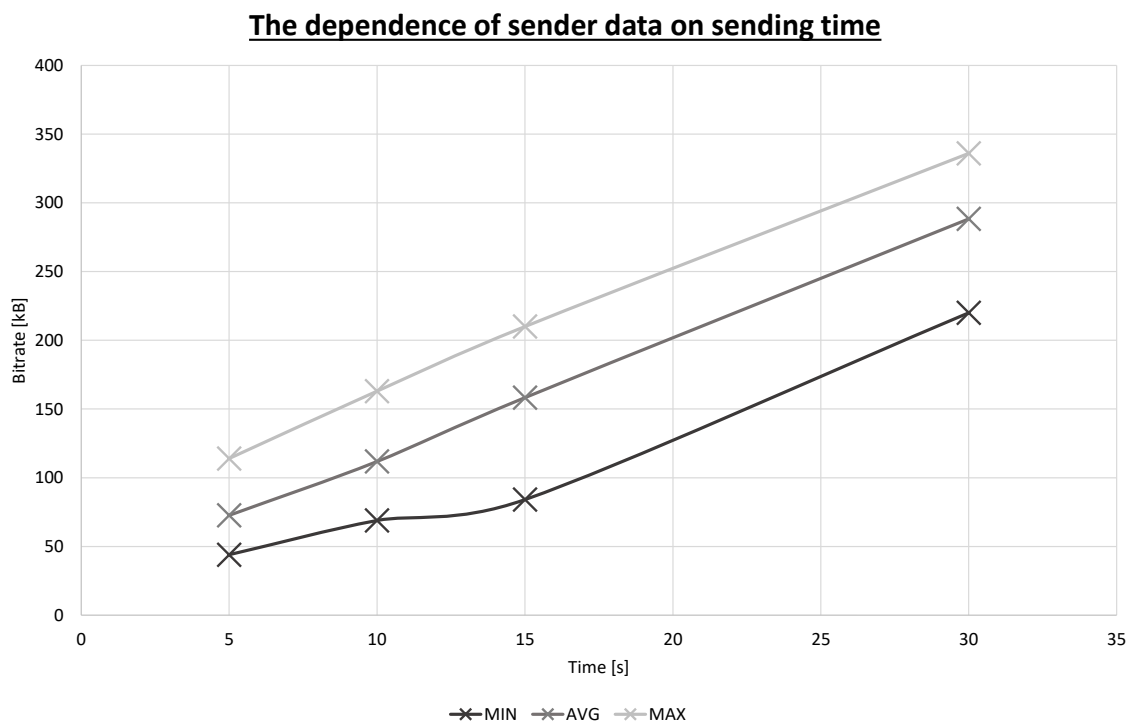


Fig. 4.5: The Dependence of Sender Data Size on Sending Time in Brno

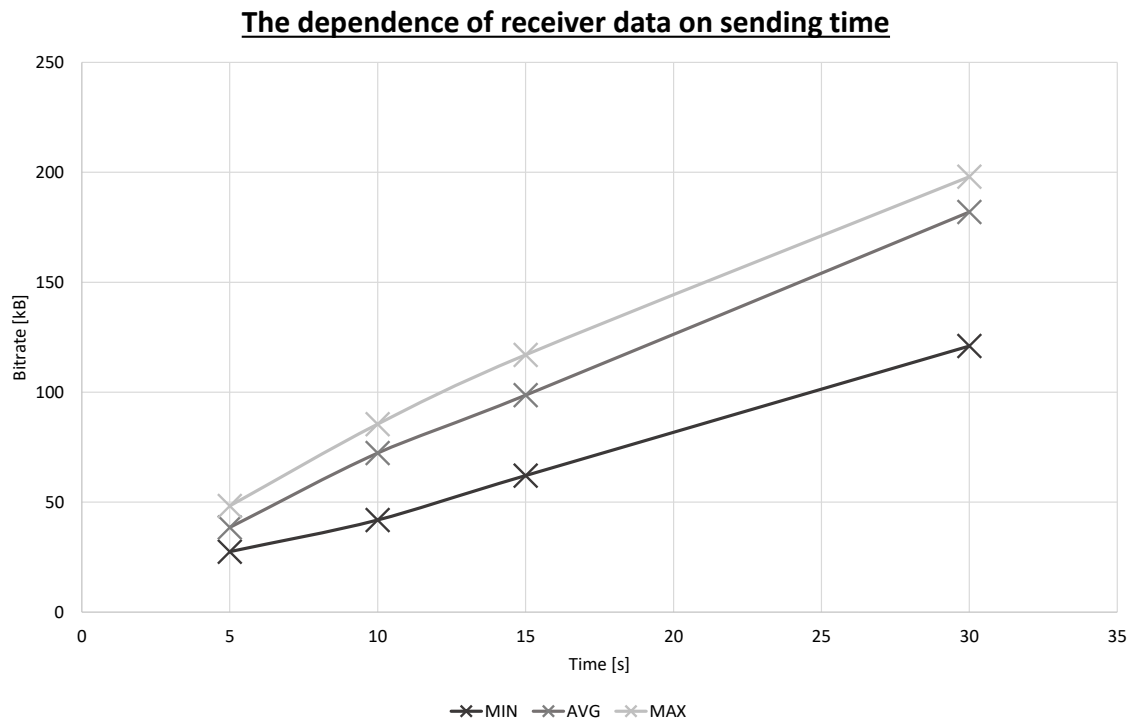


Fig. 4.6: The Dependence of Receiver Data Size on Sending Time in Brno

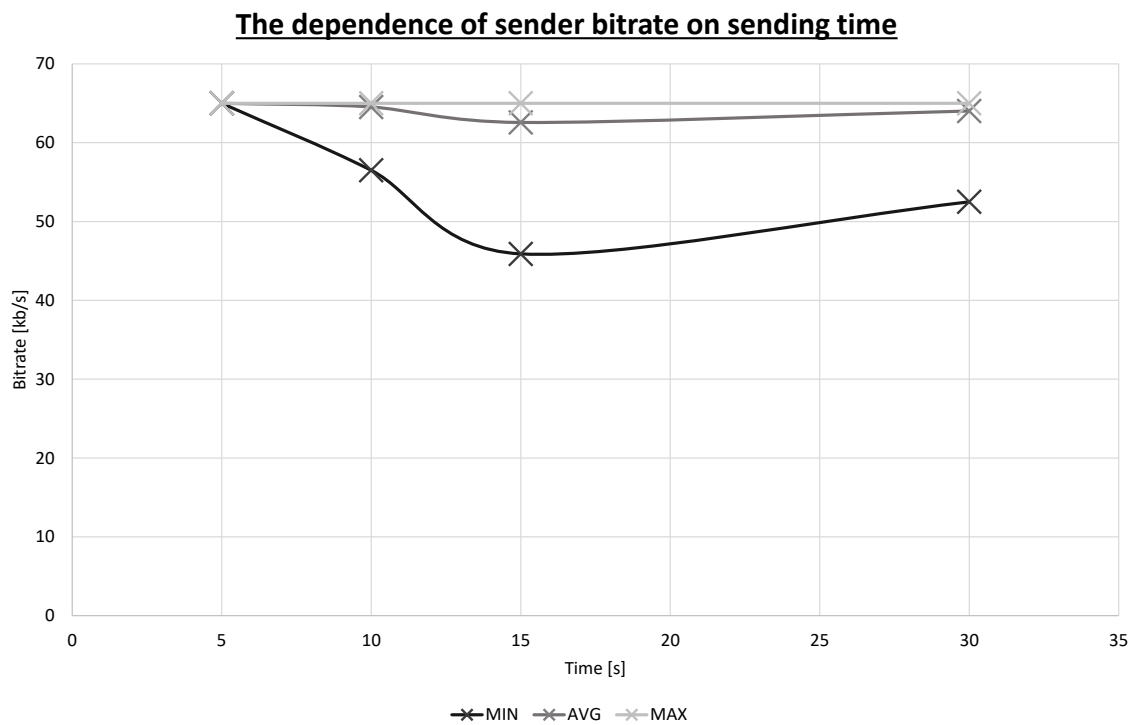


Fig. 4.7: The Dependence of Sender Bitrate on Sending Time in Brno

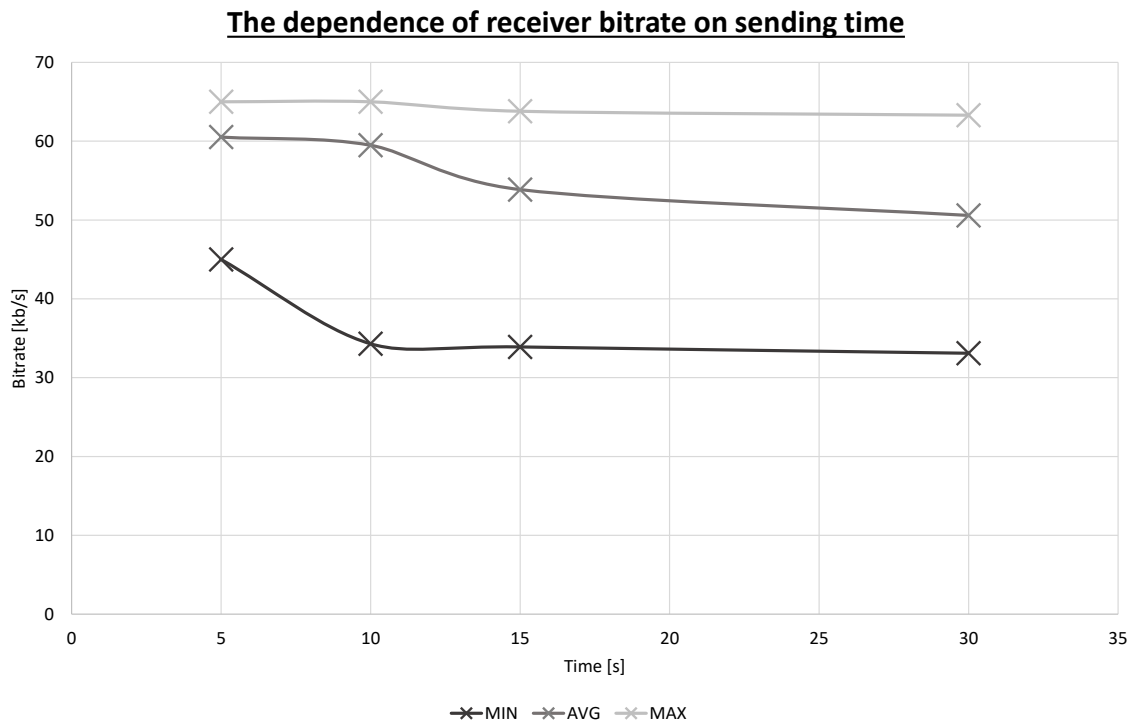


Fig. 4.8: The Dependence of Receiver Bitrate on Sending Time in Brno

Heřmanice

Maximum transported data in charts 4.9 and 4.10 slowly grows with sending time. On the contrary, the bitrates in charts 4.11 and 4.12 slowly fall in sender and receiver. It is due to a more unstable connection with more time on air.

In the figure 4.9 and 4.10, there is a measurement for 5, 10, 15, 30 s of sending time. We can see that the sender sends data in the range between 20 kB and 180 kB. On the other hand, the receiver sends data in the range between 3 kB and 90 kB.

In the figure 4.11 and 4.11 charts, there is also a measurement for 5, 10, 15, 30 s of sending time. We can see that the sender's bitrate is in the range between 7 kb/s and 60 kb/s. n the contrary, the receiver's bitrate is in the range between 1 kb/s and 43 kb/s.

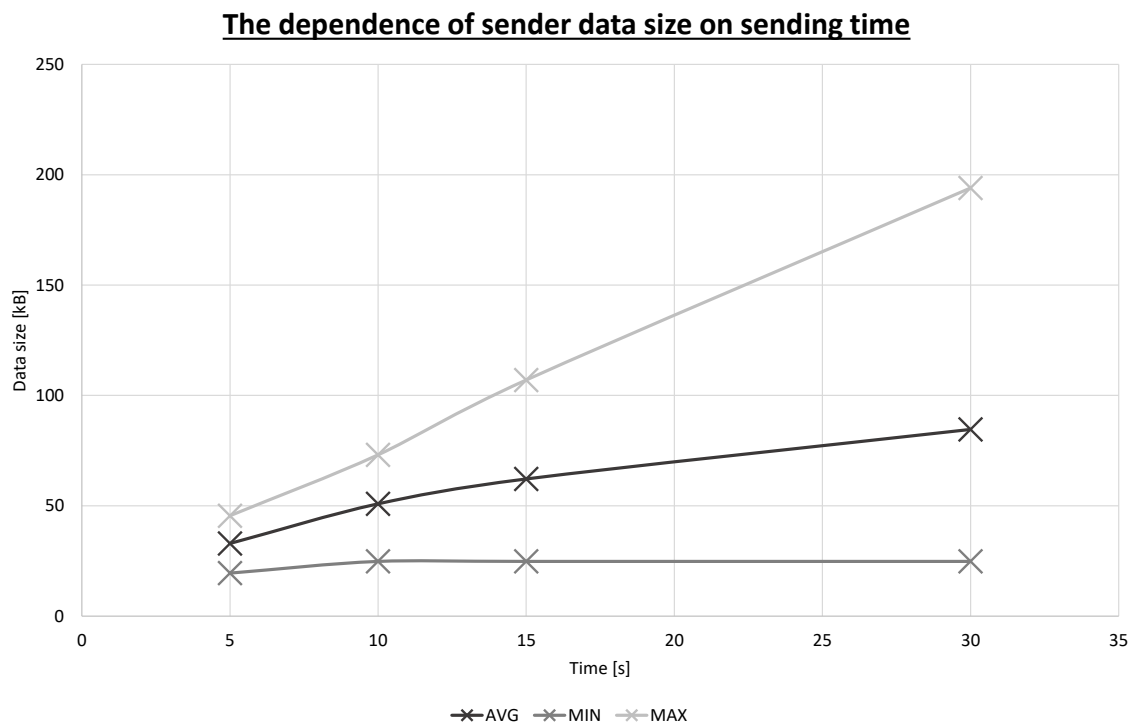


Fig. 4.9: The Dependence of Sender Data Size on Sending Time in Heřmanice

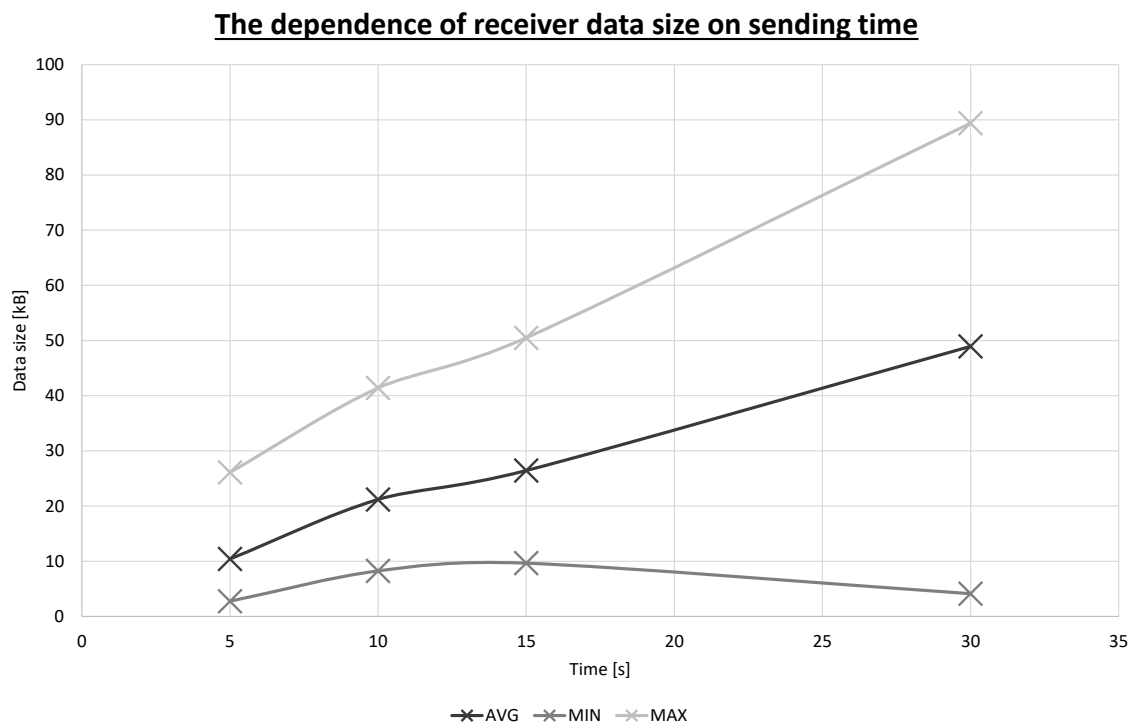


Fig. 4.10: The Dependence of Receiver Data Size on Sending Time in Heřmanice

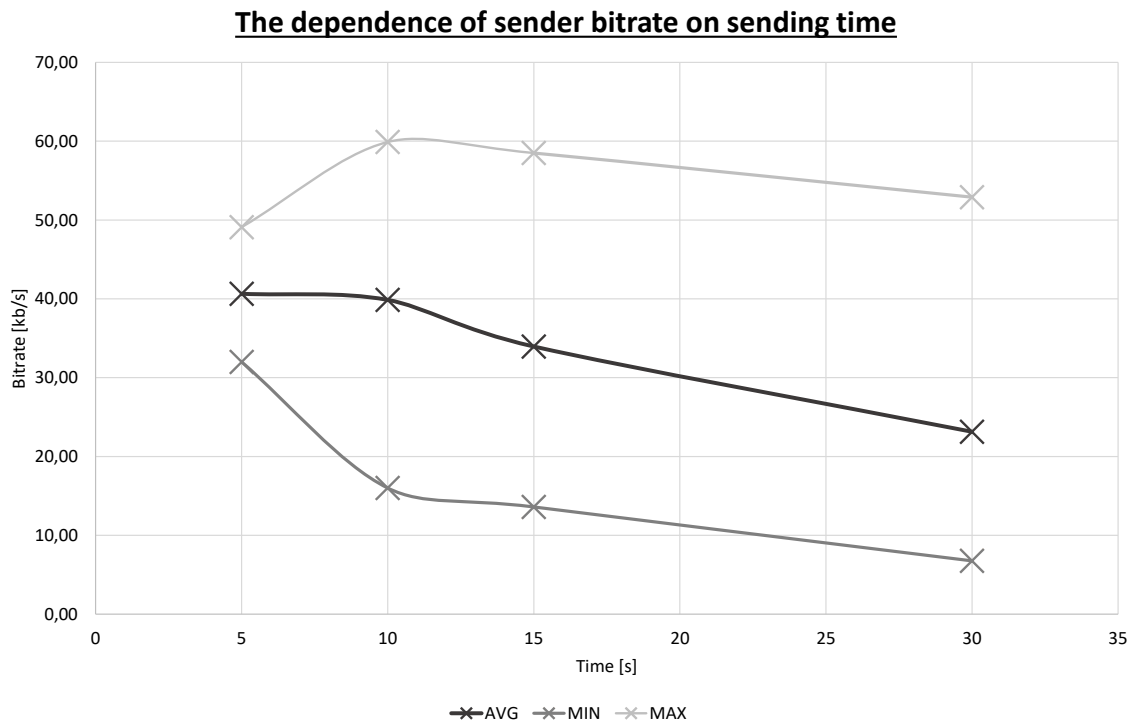


Fig. 4.11: The Dependence of Sender Bitrate on Sending Time in Heřmanice

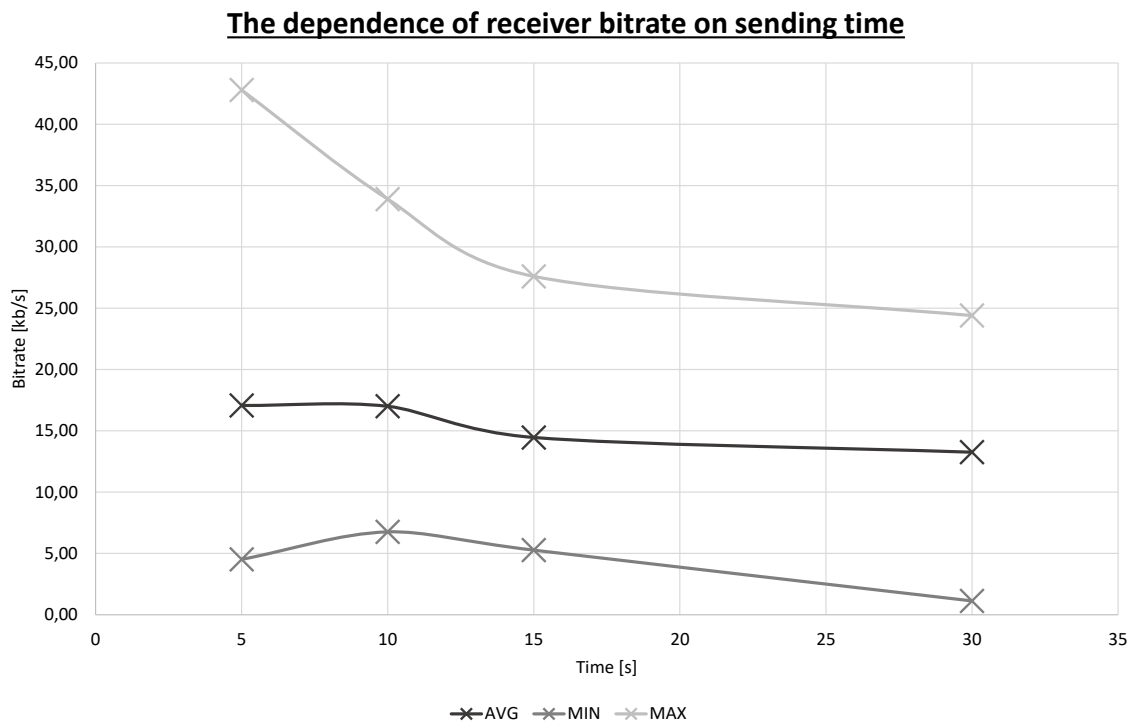


Fig. 4.12: The Dependence of Receiver Bitrate on Sending Time in Heřmanice

4.3.3 Measured Delays

The PING command was used to measure the delay between the online server and the measuring device. PING sends several packages of selected size to the selected

address and waits for the answer. After getting the answer, PING calculates packet loss and delay between the measuring device and the selected server. In chart 4.13, there is packet loss for two locations. We can be seen that Brno has minor packet loss in general, apart from Heřmanice. Brno has maximum packet loss same for 8000kB and 10000kB at 50 %. Heřmanice, on the other hand, has the maximum even more significant and that 75 % for 5000 kB size of the package.

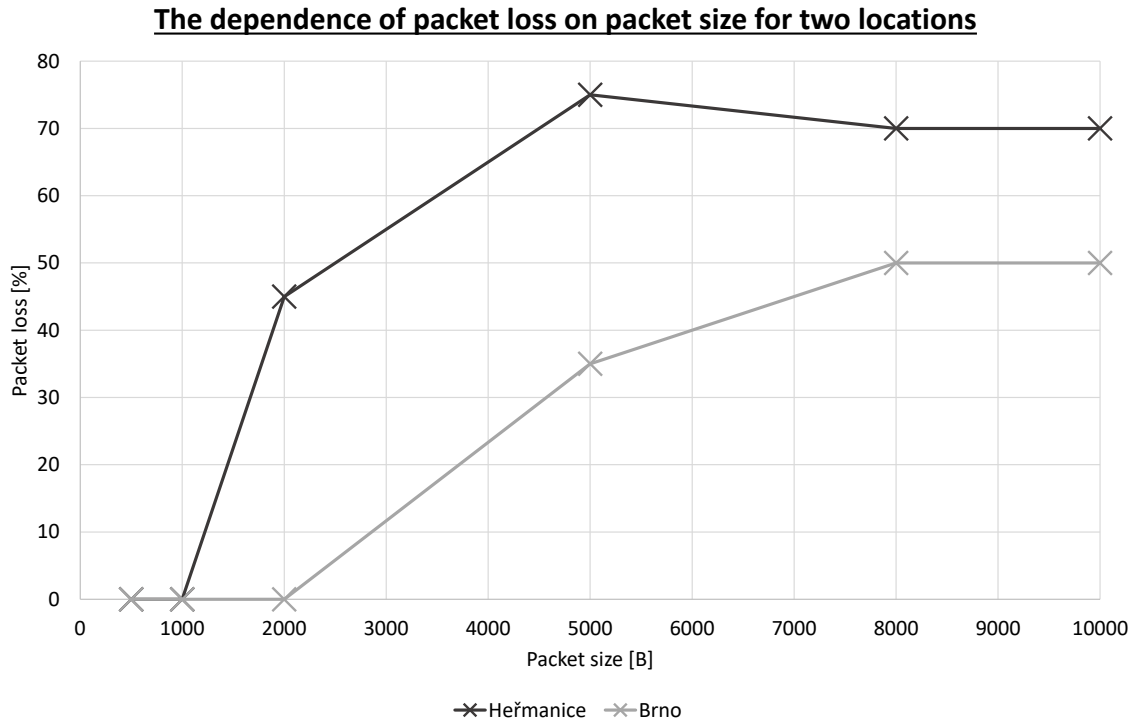


Fig. 4.13: The Dependence of Packet Loss on Packet Size for Brno and Heřmanice

Brno

From chart 4.14, we can see that the time between the first send and last receive of PING packet sizes 500kB and 5000kB is similar, and for last two packet size gets bigger. The larger delay values surprisingly have packet sizes of 8000kB. It can happen because of an unstable connection for a big packet size. Overall the values are better than in Heřmanice shown In the figure 4.15.

Heřmanice

From the graph 4.15, we can see that, as in the figure 4.14, the time between the first sending and the last receiving of a PING packet is similar for 500 kB to 5000 kB, and becomes larger for the last two packet sizes. Unlike in measuring in Brno values, larger packet size gets even more significant. In Brno, measuring time maximum is

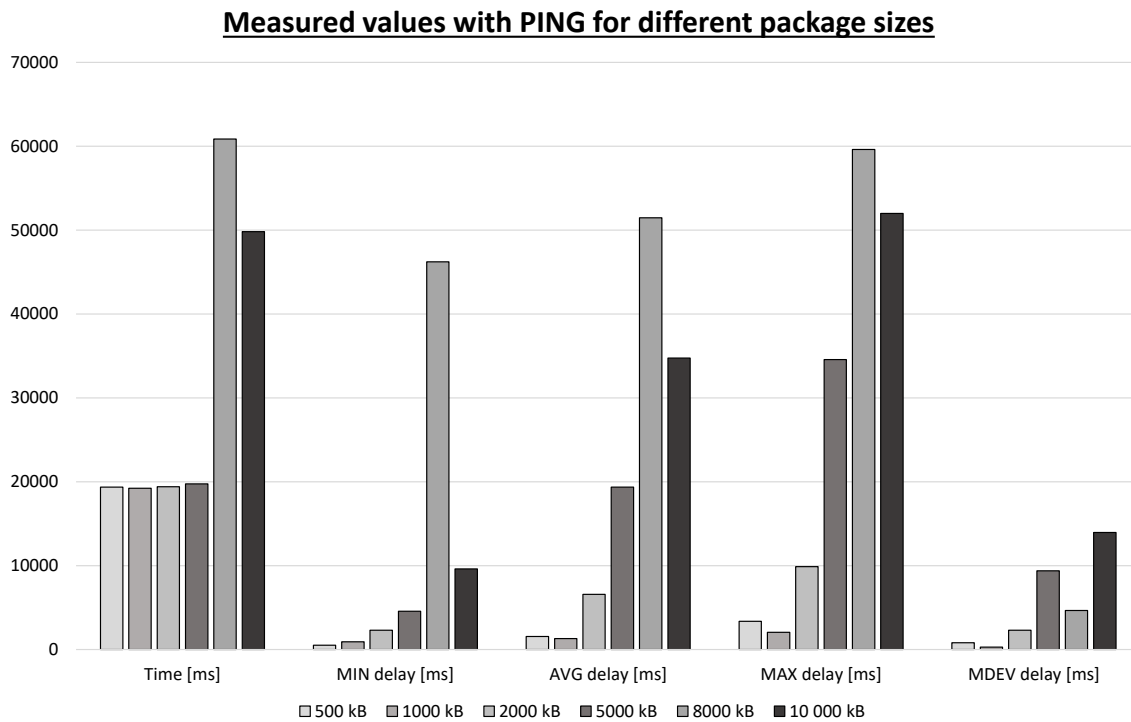


Fig. 4.14: Measured Values With Ping for Different Package Sizes in Brno

only half of what was measured in Heřmanice. The delay grew with a larger packet size similar 4.14. The only difference is that these delay values get up to 100000 ms, which is almost double Brno's values.

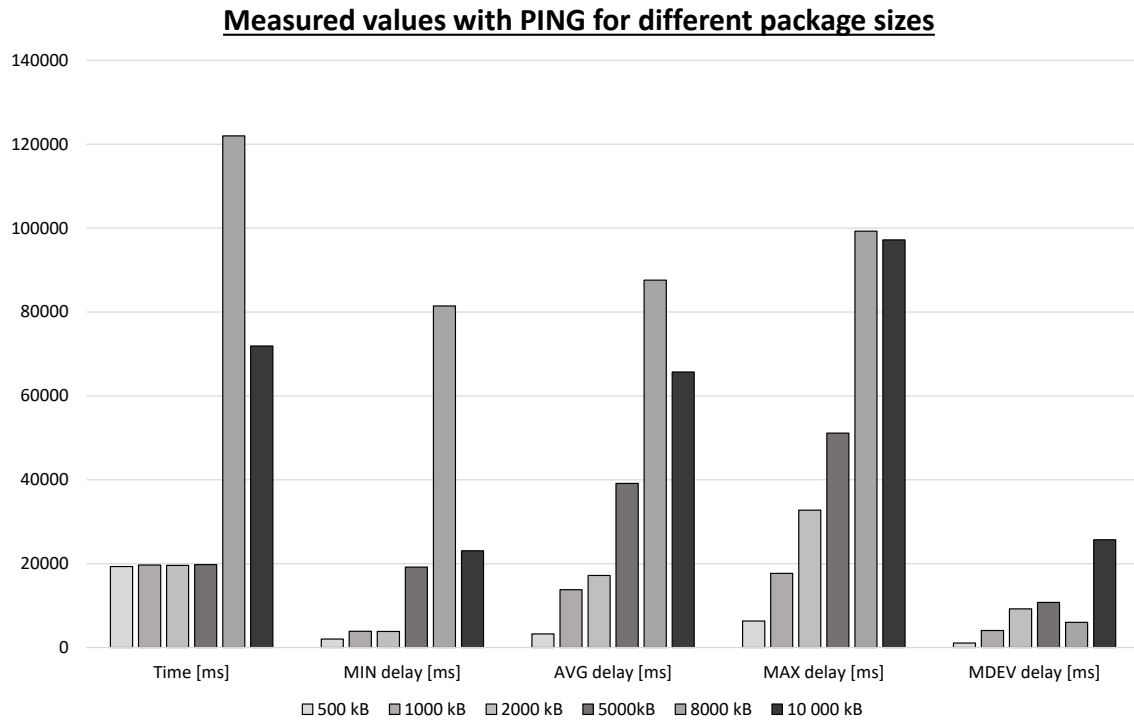


Fig. 4.15: Measured Values With Ping for Different Package Sizes in Heřmanice

4.4 Infrastructure of DLMS/COSEM Communication

RPi with BG77 was used as an NB-IOT device for testing infrastructure working like the smart meter on the server side of communication. The communication is connected through the radio-frequency NB-IoT network to the NB-IoT base station. The communication goes to the BUT server, where it is forwarded, thanks to the correct port, to the server. The server is a Linux-based platform that uses specific scripts to communicate with RPi.

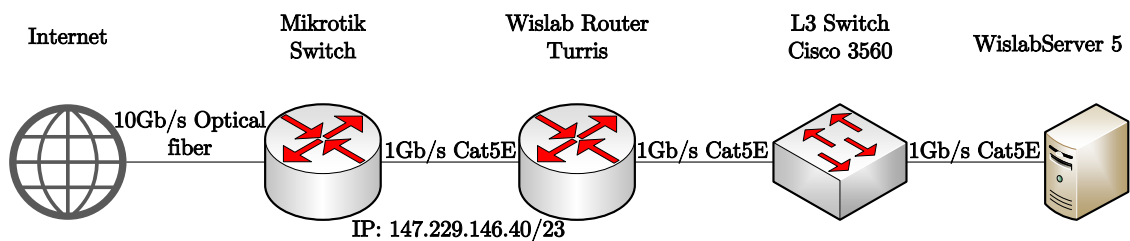


Fig. 4.16: Real Infrastructure of Connection

4.5 TLS Communication Tunnel

For creating the TLS tunnel, the Stunnel application is used. More information about the TLS tunnel is in figure 3.5. The TLC v1.2 is used to set the communication, and it uses AES encryption. AES is a symmetric block cypher with various lengths. Stunnel is software that checks the server and client's private key and certificate. As next, it makes the TLS handshake and secures all connections. It also can redirect different ports and IP addresses. The Stunnel has to be established from the server side (RPi side) of the communication because it has no public IP address. After that client knows where to connect[35]. The authentication steps between client and server:

- client initiates communication towards the server
- server requests client's certificate
- client sends encryption specifications
- server validates and responds to encryption specifications
- server sends modified encryption specifications back to the client
- mutual authentication complete

For the server part, the Stunnel configuration was chosen as follows:

- output = /etc/stunnel/stunnel.log
- cert = /etc/stunnel/230558.pem
- Cfile = /etc/stunnel/230558.pem
- client = yes
- debug = 6
- verify = 4
- [horcicka]
- accept = 127.0.0.1:65400
- connect = 147.229.146.40:65405.

For the client part, the Stunnel configuration was chosen as follows:

- output = stunnel.log
- cert = stunnel.pem
- Cfile = ca-certs.pem
- client = no
- debug = 6
- verify = 4
- [horcicka]
- accept = 192.168.85.163:65405
- connect = 127.0.0.1:65401.

4.6 Communication

The first possibility was to connect the 5G laboratory unique desk to the actual smart meter. My supervisor was disgusted with this option, and we decided not to do it like that. After that was agreed upon to create on RPi, the emulator of DLMS/COSEM data simulating the smart meter, the server part of the DLMS/COSEM connection. For that purpose was recommended to use Gurux DLMS/COSEM python library[36].

The Python library is enormous and has examples for all Clients' purposes. However, as we know, for this task, the RPi must act like a smart meter, which means as a server in DLMS/COSEM communication. When using the Gurux DLMS/COSEM library, it had a problem working through the Stunnel TLS tunnel. I could not make it work as necessary to use the specific library there. The other way was chosen for finalizing the bachelor thesis. Namely, the most thoughtful way to make it work was to create a light-verse communication between the server and the client by hand. On both sides, the code is written in Python programming language. A socket library is used to create the connection through the TLS tunnel. The whole communication between the server and client goes through a 5G NB-IoT network.

4.6.1 Client Side of the Communication

As the client side of the communication, the Linux-based server is being used. Firstly there has to be a connection from RPi because the RPi doesn't have public IP, so a connection to it isn't possible. After the connection, the client is ready to ask for specific data from the server. There was a specified Integer which represented data generated by SM and clock. Both of these data have their own OBIS code. The OBIS code is used in DLMS/COSEM communication to specify the target data the client wants to read. Client first asks for integer data, then waits for a response and requests for clock data, then waits for 15s and does it again. The primary function here takes OBIS code and formats it with other data to DLMS/COSEM format message. The DLMS/COSEM messages are sent in hexadecimal format through the TLS tunnel. The client then converts the received data to the original form and writes it out. Console output is visible in 4.17.

4.6.2 Server Side of the Communication (RPi)

As the server side of the communication, the RPi is being used. First, the server has to connect to the client. After a successful connection, the server waits for the client to request data. The client has to ask with a specified OBIS code for the server to answer. If there is a wrong OBIS code, the server won't respond. On the other

hand, if the OBIS code is correct server will send desirable data. The server can provide the integer representing the data generated by SM and clock. The server converts all its data to DLMS/COSEM message format. For this purpose, there are two main functions one for data and one for clock. Each process converges data to the hexadecimal bytes sequence and sends it through the socket communication. Console output is visible in 4.18.

```

lpwan@WislabServer5:~/230558$ python3 FINALserver.py
Listening...
Connected to ('127.0.0.1', 37326)
DATA REQUEST..
TX \x10 \x09 \xC0 \x01 \x00 \x1F \x07 \x00 \xFF
RX: \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x11 \x9b
Received data: 4507
TIME REQUEST..
TX \x10 \x09 \xC0 \x00 \x00 \x01 \x00 \x00 \xFF
RX: \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \x8c
Current time: 2023-08-09 16:54:04

DATA REQUEST..
TX \x10 \x09 \xC0 \x01 \x00 \x1F \x07 \x00 \xFF
RX: \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x3e \xc7
Received data: 16071
TIME REQUEST..
TX \x10 \x09 \xC0 \x00 \x00 \x01 \x00 \x00 \xFF
RX: \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \x9b
Current time: 2023-08-09 16:54:19

DATA REQUEST..
TX \x10 \x09 \xC0 \x01 \x00 \x1F \x07 \x00 \xFF
RX: \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x0d \xa5
Received data: 3493
TIME REQUEST..
TX \x10 \x09 \xC0 \x00 \x00 \x01 \x00 \x00 \xFF
RX: \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \xaa
Current time: 2023-08-09 16:54:34

DATA REQUEST..
TX \x10 \x09 \xC0 \x01 \x00 \x1F \x07 \x00 \xFF
RX: \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x3c \x34
Received data: 15412
TIME REQUEST..
TX \x10 \x09 \xC0 \x00 \x00 \x01 \x00 \x00 \xFF
RX: \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \xb9
Current time: 2023-08-09 16:54:49

DATA REQUEST..
TX \x10 \x09 \xC0 \x01 \x00 \x1F \x07 \x00 \xFF
RX: \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x2b \xec
Received data: 11244
TIME REQUEST..
TX \x10 \x09 \xC0 \x00 \x00 \x01 \x00 \x00 \xFF
RX: \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \xc8
Current time: 2023-08-09 16:55:04

```

Fig. 4.17: Client Console Output

```

root@VfStudenti2023cu245:~# python3 FINALClient.py
Connection to the server
RX: \x10 \x09 \xc0 \x01 \x00 \x1f \x07 \x00 \xff
DATA request
TX \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x11 \x9b
RX: \x10 \x09 \xc0 \x00 \x00 \x01 \x00 \x00 \xff
TIME request
TX \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \x8c

RX: \x10 \x09 \xc0 \x01 \x00 \x1f \x07 \x00 \xff
DATA request
TX \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x3e \xc7
RX: \x10 \x09 \xc0 \x00 \x00 \x01 \x00 \x00 \xff
TIME request
TX \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \x9b

RX: \x10 \x09 \xc0 \x01 \x00 \x1f \x07 \x00 \xff
DATA request
TX \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x0d \xa5
RX: \x10 \x09 \xc0 \x00 \x00 \x01 \x00 \x00 \xff
TIME request
TX \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \xaa

RX: \x10 \x09 \xc0 \x01 \x00 \x1f \x07 \x00 \xff
DATA request
TX \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x3c \x34
RX: \x10 \x09 \xc0 \x00 \x00 \x01 \x00 \x00 \xff
TIME request
TX \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \xb9

RX: \x10 \x09 \xc0 \x01 \x00 \x1f \x07 \x00 \xff
DATA request
TX \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x2b \xec
RX: \x10 \x09 \xc0 \x00 \x00 \x01 \x00 \x00 \xff
TIME request
TX \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \xc8

RX: \x10 \x09 \xc0 \x01 \x00 \x1f \x07 \x00 \xff
DATA request
TX \x10 \x0e \x01 \x00 \x1f \x07 \x00 \xff \x04 \x00 \x00 \x1e \x4a
RX: \x10 \x09 \xc0 \x00 \x00 \x01 \x00 \x00 \xff
TIME request
TX \x10 \x0e \x00 \x00 \x01 \x00 \x00 \xff \x04 \x64 \xd3 \xa8 \xd7

```

Fig. 4.18: Server Console Output

Conclusion

Smart metering is an essential part of the modern energy consumption system. It can decrease the need for energy so people would save money. Smart metering also helps suppliers with a more straightforward reading of meters and real-time statistics from obtained data. Intelligent monitoring can provide real-time information or alerts and ultimately save money. Most modern smart meters are capable of 5G communication on their own, but older ones need a device that does communication for them.

LPWAN (Low Power Wide Area Network) technologies are designed to work with a wide range of devices, so even when every home has a smart meter, this technology will still work well. LoRaWAN and Sigfox are in unlicensed frequency bands, and NB-IoT (Narrow-Band Internet of Things) and LTE Cat-M are in licensed frequency bands. Only devices using NB-IoT technology can be used since the LTE Cat-M does not provide broad coverage in the Czech Republic. LTE Cat-M only covers certain areas like part of BUT (Brno University of Technology). NB-IoT has extensive coverage and low battery consumption, so it is ideal for metering. Both technologies support MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), LwM2M (Lightweight Machine-to-Machine), and DLMS/COSEM (Device Language Message Specification/Companion Specification for Energy Metering) as application protocols. Due to the parameter variations, each of these protocols has advantages in different scenarios. DLMS/COSEM is most likely used for metering as it is supported by smart meters.

The first part of the practical part focused on verifying NB-IoT communication technology. Communication was verified by sending data thru UMTS & LTE EVB Kit and BG770A-GL-TE-A, which were used for connecting to the NB-IoT network. UMTS & LTE EVB Kit was used as a hardware hub for BG770A-GL-TE-A to connect with the server. Getting data to the server side has verified that the communication works. Also, communication infrastructure and used hardware were described in this chapter. The second part of the practical part focused on measuring radio properties in different locations. Iperf and Ping commands were used for data collection. All the essential measured data are processed into charts with labels and are compared to the NB-IoT boundary condition. The last practical part focused on generating and transmitting the data between server and client. The Raspberry Pi4 was used as the server side, and the Linux-based server was used as the client side. The transmission between the client and server failed with the use of the Gurux DLMS/COSEM library. So the final coding was made by hand and is much simpler. The server and client connection was made through the 5G NB-IoT network, fulfilling the request from the assigned. The written program makes a con-

nection and then sends between devices generated data in DLMS/COSEM format to simulate the behaviour of the actual smart meter.

Bibliography

- [1] Olof Liberg, Marten Sundberg, Eric Wang, Johan Bergman, and Joachim Sachs. *Cellular Internet of things: technologies, standards, and performance*. Academic Press, 2017.
- [2] Internet of things: The five types of iot, Aug 2022. URL: <https://syntegra.net/internet-of-things-the-five-types-of-iot/>.
- [3] CR Srinivasan, B Rajesh, P Saikalyan, K Premsagar, and Eadala Sarath Yadav. A review on the different types of internet of things (iot). *Journal of Advanced Research in Dynamical and Control Systems*, 11(1):154–158, 2019.
- [4] Jixuan Zheng, David Wenzhong Gao, and Li Lin. Smart meters in smart grid: An overview. In *2013 IEEE Green Technologies Conference (GreenTech)*, pages 57–64. IEEE, 2013.
- [5] Aug 2022. URL: https://en.wikipedia.org/wiki/Gas_meter.
- [6] J Halttunen. Installation effects on ultrasonic and electromagnetic flowmeters: a model-based approach. *Flow Measurement and Instrumentation*, 1(5):287–292, 1990.
- [7] URL: <https://openstax.org/books/chemistry-2e/pages/5-2-calorimetry#footnote-ref1>.
- [8] Palmetto. How electric meters work and what electric meters do, Apr 2021. URL: <https://palmetto.com/learning-center/blog/how-does-electric-meter-work-and-what-does-electric-meter-do>.
- [9] Pelion Team. Using 4g lte for iot connectivity, Oct 2022. URL: <https://pelion.com/blog/education/4g-lte-iot-connectivity/>.
- [10] Massimo Condoluci and Toktam Mahmoodi. Softwarization and virtualization in 5g mobile networks: Benefits, trends and challenges. *Computer Networks*, 146:65–84, 2018.
- [11] Bharat S Chaudhari, Marco Zennaro, and Suresh Borkar. Lpwan technologies: Emerging application characteristics, requirements, and design considerations. *Future Internet*, 12(3):46, 2020.
- [12] What is sigfox 0g technology? URL: <https://build.sigfox.com/sigfox>.
- [13] Sigfox coverage map. URL: <https://www.sigfox.com/en/coverage>.

- [14] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. Overview of cellular lpwan technologies for iot deployment: Sigfox, lorawan, and nb-iot. In *2018 ieee international conference on pervasive computing and communications workshops (percom workshops)*, pages 197–202. IEEE, 2018.
- [15] Lorawan coverage, Oct 2022. URL: <https://lorawan-alliance.org/about-lorawan/>.
- [16] Lorawan coverage, Oct 2022. URL: <https://lorawan-alliance.org/lorawan-coverage/>.
- [17] Lte cat-m infrastructure. URL: <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-LTE-M.html>.
- [18] Mobile iot deployment map, Feb 2022. URL: <https://www.gsma.com/iot/deployment-map/>.
- [19] MSc. Martin Štůsek. Research on reliable low-power wide-area communications utilizing multi-rat lpwan technologies for iot applications. URL: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=233542.
- [20] Introducing 3gpp. URL: <https://www.3gpp.org/about-us/introducing-3gpp>.
- [21] Pascal Jörke, Robert Falkenberg, and Christian Wietfeld. Power consumption analysis of nb-iot and emtc in challenging smart city environments. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2018.
- [22] Borja Martinez, Ferran Adelantado, Andrea Bartoli, and Xavier Vilajosana. Exploring the performance boundaries of nb-iot. *IEEE Internet of Things Journal*, 6(3):5702–5712, 2019.
- [23] Matthieu Kanj, Vincent Savaux, and Mathieu Le Guen. A tutorial on nb-iot physical layer design. *IEEE Communications Surveys & Tutorials*, 22(4):2408–2446, 2020.
- [24] Ing. Radim Dvořák. Vytvoření laboratorní úlohy pro technologie lpwa využívající protokol lightweight m2m (lw-m2m). URL: <https://www.vut.cz/studenti/zav-prace/detail/141376>.
- [25] Jiaying Guo, Chunxiang Gu, Xi Chen, and Fushan Wei. Model learning and model checking of ipsec implementations for internet of things. *IEEE Access*, 7:171322–171332, 2019.

- [26] Ramao Tiago Tiburski, Leonardo Albernaz Amaral, Everton de Matos, Dario FG de Azevedo, and Fabiano Hessel. Evaluating the use of tls and dtls protocols in iot middleware systems applied to e-health. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 480–485. IEEE, 2017.
- [27] Adisorn Kheaksong and Wilaiporn Lee. Packet transfer of dlms/cosem standards for smart grid. In *The 20th Asia-Pacific Conference on Communication (APCC2014)*, pages 391–396. IEEE, 2014.
- [28] Thobekile J Ngcobo and Farzad Ghayoor. An overview of dlms/cosem and g3-plc for smart metering applications. *International Journal on Smart Sensing and Intelligent Systems*, 15(1):1–15, 2022.
- [29] Dlms-overview. URL: <https://www.dlms.com/dlms-cosem/overview>.
- [30] Green Book. Dlms/cosem architecture and protocols. *Ed10. DLMS User Association*, 2020.
- [31] Blue Book. Cosem inteface classes and obis object identification system. *Zug, Switzerland: DLMS User Association*, 2017.
- [32] Lpwa bg77 cat m1/nb2, Oct 2022. URL: <https://www.quectel.com/product/lte-bg77-cat-m1-nb2>.
- [33] Raspberry pi cm4 datasheet. URL: <https://datasheets.raspberrypi.com/cm4/cm4-datasheet.pdf>.
- [34] Bg770a-gl&bg95xa-gl tcp/ip application note, Jul 2021. URL: https://www.quectel.com/wp-content/uploads/2021/08/Quectel_BG770A-GLBG95xA-GL_TCPIP_Application_Note_V1.0.pdf.
- [35] Stunnel information. URL: <https://www.stunnel.org/index.html>.
- [36] Gurux dlms for python. URL: <https://www.gurux.fi/node/13678>.
- [37] Wael Ayoub, Abed Ellatif Samhat, Fabienne Nouvel, Mohamad Mroue, and Jean-Christophe Prévotet. Internet of mobile things: Overview of lorawan, dash7, and nb-iot in lpwans standards and supported mobility. *IEEE Communications Surveys & Tutorials*, 21(2):1561–1581, 2018.
- [38] Quectel, Dec 2017. URL: https://www.quectel.com/wp-content/uploads/pdfupload/Quectel_UMTS%20%20LTE_EVB_User_Guide_V2.1.pdf.

- [39] Quectel, Dec 2021. URL: https://www.quectel.com/wp-content/uploads/2021/09/Quectel_BG770A-GL_LPWA_Specification_V1.0.pdf.
- [40] Gurux dlms simulator. URL: <https://www.gurux.fi/Gurux.DLMS.Simulator>.
- [41] Gurux dlms director. URL: <https://www.gurux.fi/Download>.

Symbols and abbreviations

IoT	Internet of Things
CIoT	Cellular Internet of Things
H2H	Human-to-Human
M2M	Machine-to-Machine
mMTC	massive Machine-Type Communication
2G	2. Generation of Mobile Networks
3G	3. Generation of Mobile Networks
4G	4. Generation of Mobile Networks
5G	5. Generation of Mobile Networks
LPWAN	Low-Power Wide Area Networks
LoRaWAN	Long-Range Wide Area Networks
PSM	Power Saving Mode
eDRX	extended Discontinuous Reception
DLMS/COSEM	Device Language Message Specification/Companion Specification for Energy Metering
YAT	Yet Another Terminal
MQTT	Message Queuing Telemetry Transport
CoAP	Constrained Application Protocol
LwM2M	Lightweight Machine-to-Machine
IPSec	Internet Protocol Security
TLS	Transport Layer Security
TA	Time Advance
ToA	Time of Arrival
BTS	Base Transceiver Station

RPi CM4	Raspberry Pi Compute module 4
PSS and SSS	Primary Synchronization Signal and Secondary Synchronization Signal
RSS	Resynchronization Signal
CRS	The Cell-Specific Reference Signal
3GPP	3rd Generation Partnership Project
NB-IoT	Narrow-Band Internet of Things
NPSS/NSSS	Narrowband Primary Synchronization Signal and Narrowband Secondary Synchronization Signal
NPBCH	Narrowband Physical Broadcast Channel
NPDCCH	Narrowband Physical Downlink Control Channel
NPDSCH	Narrowband Physical Downlink Shared Channel
NPRACH	Narrowband Physical Random Access Channel
NPUSCH	Narrowband Physical Uplink Shared Channel
DMRS	Demodulation Reference Signal
PRS	Positioning Reference signal
PBCH	Physical Broadcast Channel
MWUS	MTC Wake-Up Signal
MPDCCH	MTC Physical Downlink Control Channel
PDSCH	Physical Downlink Shared Channel
PRACH	Physical Random-Access Channel
DMRS	Demodulation Reference Signal
RS	Reference Signal
SRS	Sounding Reference Signal
PUSCH	Physical Uplink Shared Channel
PUCCH	Physical Uplink Control Channel

AAs	Application Associations
BUT	Brno University of Technology

5 Examples of Used Python Codes

Listing 5.1: Example of Generating Read Request in Python.

```
#Generating read request
def generate_dlms_cosem_read_request(obis_code):
    header = b'\x10'
    read_request_code = b'\xC0'
    numbers = obis_code.split('.')
    hexo = [hex(int(nus)) for nus in numbers]
    obis = bytes.fromhex(''.join([num[2:].zfill(2) for num in
        hexo]))
    length = len(obis) + 3

    # Construct the read request message
    read_request_message = header + bytes([length]) +
        read_request_code + obis
    formatted_message = '\x'.join(['\x{:02X}'.format(byte)
        for byte in read_request_message])
    return formatted_message
```

Listing 5.2: Example of Generating Data in Python.

```
# Generating DLMS/COSEM message
def generate_dlms_cosem_message(obis_code, int_value):
    header = b'\x10'
    numbers = obis_code.split('.')
    hexo = [hex(int(nus)) for nus in numbers]
    obis = bytes.fromhex(''.join([num[2:].zfill(2) for num in
        hexo]))

    int_bytes = int_value.to_bytes(4, byteorder='big') #
        Convert int_value to 4-byte big-endian bytes

    length = len(obis) + len(int_bytes) + 4
    message = header + bytes([length]) + obis + bytes([len(
        int_bytes)]) + int_bytes
    return format_bytes_to_hex(message)
```

Listing 5.3: Example of Generating Current Time in Python.

```
# Generating current time 1
def generate_dlms_cosem_time(obis_code, time_value): 2
    header = b'\x10' 3
    numbers = obis_code.split('.') 4
    hexo = [hex(int(nus)) for nus in numbers] 5
    obis = bytes.fromhex(''.join([num[2:].zfill(2) for num in 6
        hexo])) 7

    # Convert time to Unix timestamp 8
    timestamp = int(time_value.timestamp()) 9

    # Convert Unix timestamp to byte string 10
    time_bytes = timestamp.to_bytes(4, byteorder='big') 11

    length = len(obis) + len(time_bytes) + 4 12
    message = header + bytes([length]) + obis + bytes([len( 13
        time_bytes)]) + time_bytes 14
    return format_bytes_to_hex(message) 15
16
```

6 Content of the electronic attachment

```
/.....root of the attached archive
├── pdf.....PDFs (generate them in the information system)
│   ├── BP-assignment.pdf
│   ├── cover-example.pdf
│   └── titlepage-example.pdf
├── pict.....other graphic files
│   ├── 3GPP technologies evolution.pdf
│   ├── AT-flowchart.pdf
│   ├── B-BitSender.pdf
│   ├── Laboratory desk Kit.jpg
│   ├── B-DataReceiver.pdf
│   └── ...
├── codes.....Written Python codes
│   ├── Client.py
│   └── Server(Rpi).py
└── Bachelor's thesis Horčíčka.....Bachelor's thesis in pdf
```