AALBORG UNIVERSITET

Department Of Electronic Systems

Wireless M-Bus based eXtremely Low Power protocol for wireless communication with water meters

Group 1000 NDS

Dimo Donev

31. May 2012
Supervisor: Neeli Prasad

AALBORG UNIVERSITET

Department of Electronic Systems I-8
Fredrik Bajers Vej 7
9220 Aalborg Øst
http://www.es.aau.dk/

Title:
Wireless M-Bus based eXtremely Low Power protocol for wireless communication with water meters

Theme:
Power efficient protocols

Project Period:
Spring 2012

Project Group:
1001

Member:

Dimo Donev

Supervisor:
Neeli Prasad

Number of copies printed and pages: 3, 68

Finished the 31 of May 2012.

Synopsis:

This report investigates the possibility of creating an eXtreme Low Power protocol for wireless communication with water meters. It is based on Wireless M-Bus protocol and focuses on modifying it in order to achieve maximum power efficiency in the meter side. The work is concentrated on investigations about the possibility of reducing the transmitting packet up to a bare minimum and the ways it can be recovered after its reception.
A motivation for the problem and overview of Wireless M-Bus are presented. It is investigated the requirements and limitations for communication with water meters. Simulation tests, conclusions and proposal for operating under certain conditions are given. At the end, overview of the work is done and future tasks for developing and improving the current work are suggested.

Contents

Acknowledgements

I would like to express my gratitude to:

- Rasmus Melchior Jacobsen, an industrial PhD student in Aalborg University, who provided some real world information about using of Wireless M-Bus.
- Miitors for the topic of this report and the provided information and collaboration.
- Neeli Prasad and Rasmus Hjorth Nielsen from CTIF for the supervision.
- Jeremie Pierre Gay, formal Aalborg University student and current CEO of Create It Real Aps, for his assistance and help in the developing of the simulation script.

Chapter 1

Introduction

Preserving resources is getting one of the most important issues for the modern world. Together with the requirements for security and reliability, lowering of consumption and costs of the utilities and devices is main focus for most of the manufacturers and users.

1.1 Being "Smart"

With the development of the communication technologies, lately it happens more and more often to talk about wireless sensor networks,"smart" homes, "smart" buildings, "smart" cities, etc. One of the ideas behind these terms is a number of sensors, meters and utilities connected together into a network keeping track of certain parameters in the area and based on some criteria execute commands to the appliances connected in the network. A short example of "smart" home is shown on figure 1.1.

A possible scenario:

- Detectors for smoke and gas inform the owners if there is presence of such;
- When the door is opened, the lights are turned on automatically;
- The sprinkler in the garden is turned on in particular time intervals or when the soil's moisture is measured under a certain level;
- If the water consumption increase rapidly the owners are informed for possible water leaks;
- If the owners are out, they can access their house through internet and on the way home can turn on the heaters and start the coffee machine so they can enjoy a warm beverage in the moment they step into.

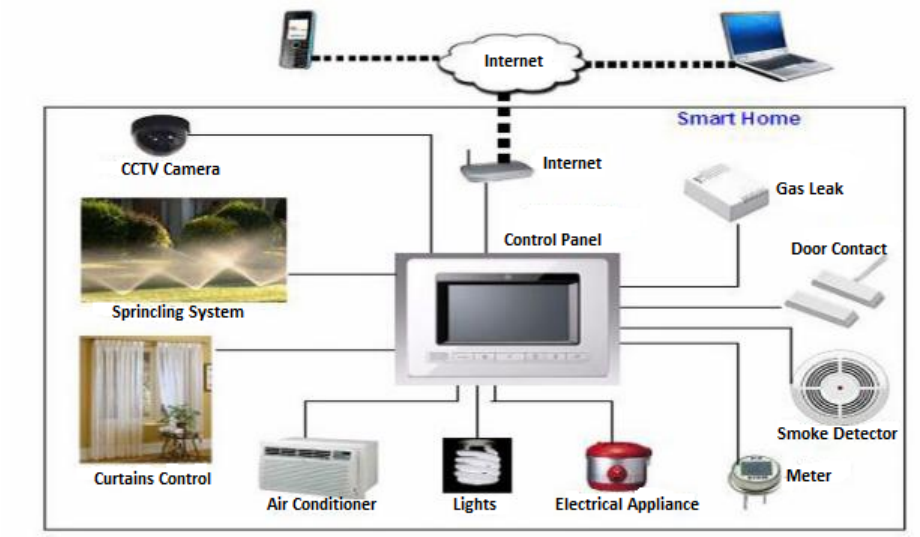

Figure 1.1: Smart Home example

1.2 AMR and AMI

Automatic Meter Reading (AMR) is a major part of "being smart". As its name implies, in its basis AMR is a technology that provides the capability of automatic data collecting from different meters. But since the technologies used for AMR provide a lot of opportunities, now this term includes also event alarms, data storage and data transfer to a base station. It gives the possibility of real time tracking of consumption, more accurate billing, and including the meters as part of bigger systems for automation and control. The existence of AMR has led to the implementation of *Advanced Metering Infrastructure* (AMI) - "systems that measure, collect and analyse energy usage, and communicate with metering devices such as electricity meters, gas meters, heat meters, and water meters, either on request or on a schedule. These systems include hardware, software, communications, consumer energy displays and controllers, customer associated systems, Meter Data Management (MDM) software, and supplier business systems"[13]. Basically what AMI includes in its meaning is AMR with the possibility of establishing two-way communication with the meters.

There are different wired and wireless technologies that Automatic Meter Reading can use - Power Line Communications (PLC), ZigBee, Bluetooth, WiFi, GPRS, M-Bus, Wireless M-Bus, Z-wave, etc. Which one will be used depends on number of factors like application, environment, metering system, budget, scalability of the system, etc. In many cases a mixture between some these technologies can be exploited.

Currently, the most frequently mentioned technology for implementing

AMR and AMI is **ZigBee** due to its energy efficient and flexible routing protocols[6]. In 2007 in Gøteborg, Sweden started a project for deploying ZigBee network, connecting all the meters in the whole city [2]. On fig. 1.2 is shown the general idea behind this project. The ZigBee equipped meters send the data to the corresponding concentrators and then the concentrators route the data to the central system. In some parts of the network where the ZigBee based communication is considered as not suitable, different communication technologies like RF, optical fibre, GPRS, etc are used.

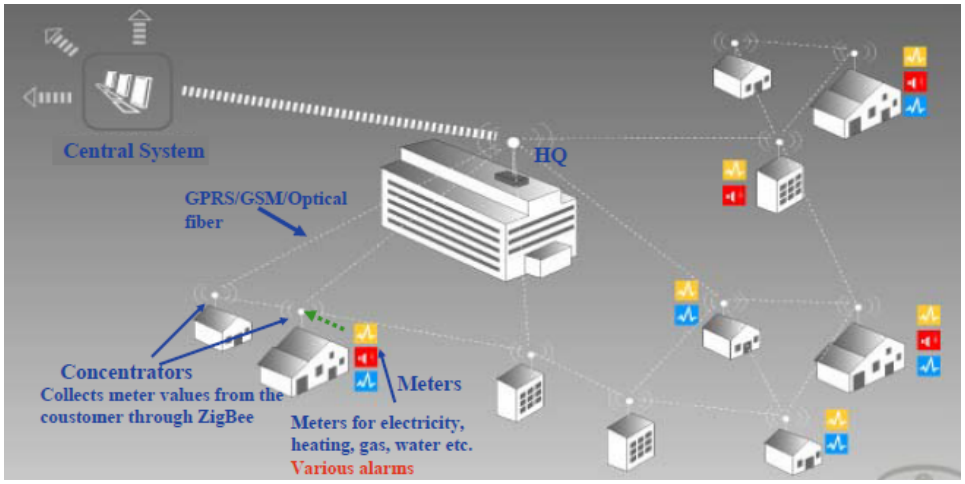

Figure 1.2: ZigBee based network for development of AMI [2]

On the other hand, currently in Europe another technology, developed specially for very power efficient smart metering and AMI applications, is getting widely accepted - Wireless M-Bus[11]. Developed as RF alternative of M-Bus [1], its initial idea is connecting all the meters inside the home and the home area (fig. 1.3). Once the data reach the data collector (concentrator), which in the general case can be also an electricity meter, it is processed and sent to a desired destination for home application or as a part of a bigger network using PLC, Bluetooth, W-LAN, GPRS etc. (depending on the application).

1.3 Seeing a Problem?

All those sensors and meters require an energy source for their functionality. In many cases connecting them to the mains is not possible so alternatives are required. Despite the rapid increase in the development of the technologies for energy scavenging, the batteries are still the main alternative energy source. However, they are expensive, environment unfriendly and, as can be seen on fig. 1.4, the slowest technology from a capacity growth (energy density) point of view. For this reason, it is very important that

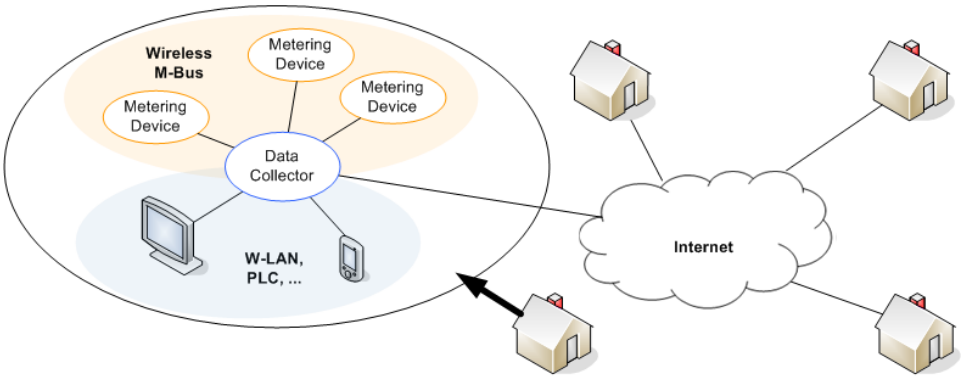

Figure 1.3: Wireless M-Bus concept

these devices have very low energy consumption and use very power efficient communication protocols for the data exchange between each other.

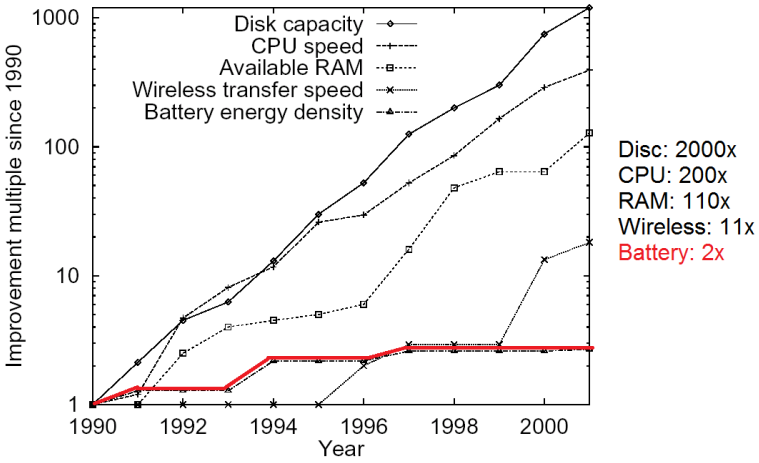

Figure 1.4: Moore's law for different units

The necessity of good protocols increases even more when the devices use radio frequency communication due to number of reasons like the possibility of interference between them, security issues and also the power consumption of the transmitter/receiver which in some cases exceeds the self consumption of the devices thousands of times.

1.4 Water meters and their place in the world

Water meter is a device whose main function is to collect information of the water consumption over a period of time in certain region/location. It is in use for everybody who needs to keep track of the water float, therefore

water consumption. Water meters are important components for the water providing companies and their customers due to number of reasons. They allow the company to charge customers for the amount of used water; monitor the water sent to the distribution system; detect leaks in the system. On the other hand, the customers can keep track of their water consumption which gives them the opportunity to control their bills, detect leaks at their location.

In the countries with a developed water providing infrastructure, every household, office building, store, etc. with access to water-main have at least one water meter. On fig. 1.5 is presented statistics for the number of households in Denmark for the period of 10 years (from 1995 to 2004). Also, keeping in mind that the population on Earth and the average standard of life are increasing, the market for these devices is huge and will keep growing.

Year	1995	1996	1997	1998	1999
Households	2 357 479	2 374 055	2 391 547	2 407 010	2 423 208
Year	2000	2001	2002	2003	2004
Households	2 434 112	2 444 467	2 456 106	2 466 693	2 480 879

Figure 1.5: Households in Denmark for the period 1995 - 2004 [3]

Water meters are not new for the market. They exist for many years and there are many companies that work in this field, therefore there are plenty of solutions for measuring and reading the water consumption.

As part of the modern world, with the progress of the technology the water meters and the way they interact with the surrounding have evolved a lot. Starting as a stand alone pure mechanical devices, during the years they have become more accurate, complex and with different capabilities allowing them to be a part of AMR/AMI, providing additional information and services for the water providers and their customers. Of course, these advantages have price - the meters are electronic, which leads to requirement of energy source.

1.5 Water meter reading methods [8]

Based on the type of water meter, there exist different methods for its reading by the water provider:

Eyeball - the traditional method in which there is a dial that shows the water consumption and a person has to reach the water meter and write down the information manually. This method requires people working full time to walk around from door to door and disturb company's clients. This leads to complications if there is nobody to open the door for the

employee. Also, the meters have to be located in places with relatively easy access. This is labor-intensive and expensive method and for this reason the information is collected very rarely (once a month or less).

Walk-by - the meter is connected with wires to a device located outside the building or is equipped with radio transmitter. It still requires a physical visit by a collector with a handheld device for collecting the data, but eliminates the problems like lack of access to the water meter. The frequency of collecting the data is the same as the Eyeball method.

Drive-by - the meter is equipped with radio transmitter. The receiver is portable, placed in a vehicle, and the information is gathered wirelessly when the reader is in the coverage of the transmitter. This method is used in rural areas, where implementation of a fixed network (see below) is not cost effective. Major problem is the energy efficiency, since the meter is supposed to be awake very often, while the receiver "shows up" very rarely. Solution for the problem can be if the meter awakes only in certain times when it is expected that the receiver will be in the area, but this requires very good time scheduling.

Fixed network - the meters are connected to a base station, repeater or to each other establishing a fixed network. This provides automatic reading of the water meter which can be practically continuous (depending on the implementation and requirements).

1.6 Motivation

Usually water meters are located in basements, closets or meter pits. These places have something in common - most probably there is no access to the mains. This leads to usage of battery for supplying the device.

Let's observe the following case: A water provider company uses battery supplied water meters for their utilities. This company supplies a whole city with water and has to equip all the households, offices, factories, etc. in the city with water meters. According to [3] for year 2004 in city of Aalborg (Denmark) only the households were 80198. Including the rest of the places where water meter is required, their number of the devices in the city can easily exceed 100000. If we assume that each meter has to have its battery changed once every 2 years and the battery discharge of the meters is equally spread in time, for 1 year the company will have to change $\frac{100000}{2 \cdot 365} \approx 137$ batteries daily, which leads to necessity of having people working full day only for changing batteries, big expenses and enormous pollution of the environment.

The current master thesis is based on collaboration with **Miitors** (www.miitors.com) - company located in Aarhus, Denmark. Their business model is develop-

ment of ultra precise and energy efficient ultrasonic water meters. Company's mission is providing state of the art water meter technology to meter manufacturers. The first product to be released is W-Miitor - battery powered water meter with Wireless M-Bus communication interface. It is estimated that the device will be capable of working for more than 10 years with one small battery.

Since the radio communication is the biggest energy drainer, reducing the time in which the radio module of the device is on will increase the battery life significantly. For this reason, from the company are interested in investigating the possibilities for implementing an eXtreme Low Power protocol (XLP), reducing the information the meter sends up to the bare minimum. Though, the reduction has to be such, that when received, with additional help from a Database containing a priori information, the message should still be possible to be extracted into a complete Wireless M-Bus datagram for further processing (if needed) (see fig. 1.6).

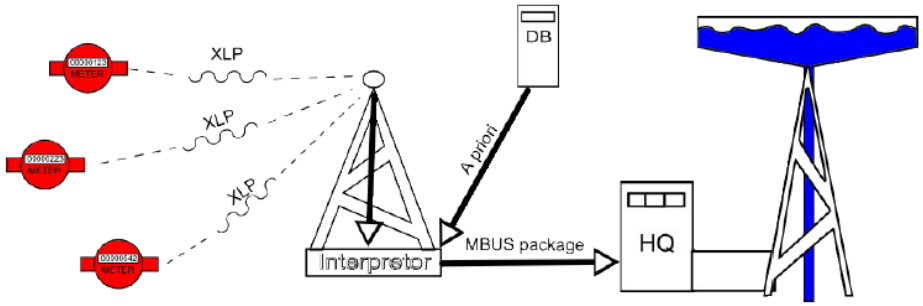

Figure 1.6: Diagram of the project statement

Chapter 2

M-Bus

M-Bus (Meter-Bus) is a standard developed in Europe for remote reading of meters for gas, water, heat and electricity. It is described by the European Norm (EN 13757) and defines the communication between meters and a data collector. Initially its interface is made for Power Line Communication (PLC), but lately it is developed a wireless communication standard, called *Wireless M-Bus*. The EN specification is divided into five parts, defining different aspects of the standard:

EN 13757-1 - *Data exchange*[4]. Describes the basic communication between meters and a data collector. It provides an overview of the communication system.

EN 13757-2 - *Physical and link layer*[5]. Describes the physical and the link layers for wired connected system.

EN 13757-3 - *Dedicated application layer*[10]. Describes the application layer. It is dedicated to vendor's

EN 13757-4 - *Wireless meter readout*[11]. Specifies the wireless communication of M-Bus. It describes the Physical and the Data link layers. Corresponds to 13757-2 for wired connection.

EN 13757-5 - *Relaying*[12]. Provides different proposals for routing the meter data in order to overcome the limited range problems between the meters and the data collectors. It is a new part of the standard and is still under development.

M-Bus uses the 3-layer IEC model. With respect to the OSI model, the standard is compatible but specifies only the Physical, Data Link and Application layers (see table 2.1).

On fig. 2.1 is shown the architecture of a M-Bus system and the dedicated layers. It should be noted that the Control and Security layers are not formal parts of M-Bus. The Control is defined by combination of specific

Layer 7	Application Layer (EN 13757-3)
Layer 2	Data Link Layer (EN 13757-2 or EN 13757-4)
Layer 1	Physical Layer (EN 13757-2 or EN 13757-4)

Table 2.1: OSI layers and M-Bus

bytes from Data Link and Application layers to specify the message exchange between the M-Bus device and the collector/concentrator. Up to date, all the other protocol layers are implemented into the Application layer. This includes also the networking layer (specified in 13757-5). The reason for this is that the standard is developed for residential point-to-point or star topology networks where on the one side are low-cost/low-power M-Bus metering devices, and on the other - data collectors or gateways (usually combined with electricity meter) with higher performance. Since the required routing protocols are still under development, mesh network topology is not available yet.

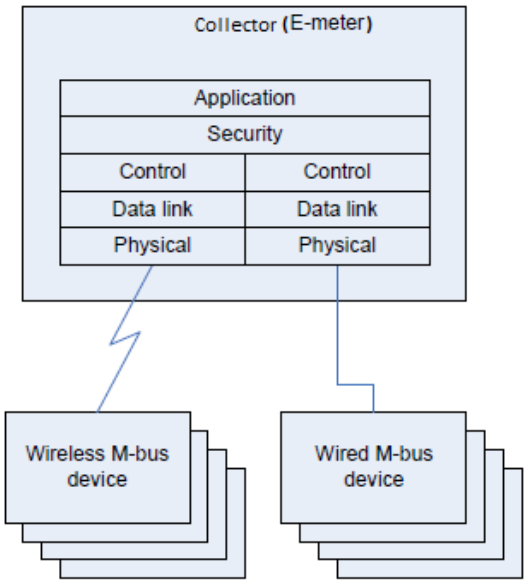

Figure 2.1: M-Bus Architecture

2.1 Wireless M-Bus - Physical Layer

Wireless M-Bus (wM-Bus) is a standard for wireless communication between metering devices and data collectors. It operates in the European ISM frequency bands - 169, 433 and $868 MHz$. The standard itself is separated in six

different operating modes (described in details in [11]), which define several different ways of exchanging data between the communicating devices.

- Mode S - Stationary
- Mode T - frequent Transmit
- Mode R2 - frequent Receive
- Mode C - Compact
- Mode N - Narrowband VHF
- Mode F - Frequent receive and transmit

Modes S, T, R2 and C use $868MHz$ frequency band, Mode N uses $169MHz$ and mode F - $433MHz$

2.1.1 Mode S

- S1 - A one way communication. The meter transmits its data without taking care if it will be received and returns immediately in power-save mode. The chiprate is $32.768kcps$, Manchester encoded with a long header. The chiprate and the encoding lead to data transfer speed $Rb = 1/2 \cdot 32.768 = 16.384kbps$ [11]. The meter sends it's data several times a day. Due to the long header, this mode is suitable for battery supplied receiver (collector).
- S1-m - This mode is the same as S1, but with a short header. For this reason it requires continuously enabled receiver.
- S2 - Bidirectional with the same parameters as S1. The header can be long or short.

2.1.2 Mode T

- T1 - One way communication. Chiprate is $100kcps$. $3to6$ encoding with short header ($Rb = 2/3 \cdot 100kcps = 66.7kbps$) [11]. Transmits with very short data bursts (3 to $8mS$) every few seconds. Suitable for walk-by and/or drive-by readout. The transmission requires at least meter ID and meter value, sent periodically.
- T2 - Two way communication. Meter-to-collector communication is as in T1, but collector-to-meter direction the chiprate is $32.768kcps$, Manchester encoded (as in S2) ($Rb = 16.384kbs$). The meter unit transmits billing data just like in T1, but after the transmission opens a short reception window for $3mS$ in case the collector has some commands or firmware information that is intended for the meter. In appendix B can be seen a detailed description of T2-mode communication scenario.

2.1.3 Mode R2

Unlike the other modes, in this one the meter does not send spontaneously data. It wakes up periodically in Rx mode and waits for a wake-up signal from the receiver. If there is no such, the meter returns in sleep mode again. In case of wake-up signal, a 2 way communication is established. Chiprate is $4.8kcps$, Manchester encoding with medium size of the header ($Rb = 2.4kbps$). Optionally it may have up to 10 frequency channels with high precision for frequency division multiplexing.

2.1.4 More C

- C1 - Its operation is like T1, but with different data encoding - NRZ. Chiprate is $100kcps$ and short header. Since for NRZ every chip sends one bit of data, $Rb = 100kbps$.

- C2 - Operates like T2. The difference is the $other-to-meter$ chiprate - $50kcps$ with NRZ encoding ($Rb = 50kbps$).

2.1.5 Mode N

This mode operates on $169MHz$ with chiprate 2.4 or $4.8kbps$ with NRZ encoding. N2g sub-mode has chiprate $16.2kcps$ but the signal is $4GFSK$ modulated, which corresponds to $Rb = 32.4kbps$. N mode is recommended for long range communication with a stationary receiver. There are several sub-modes for one-way and two-way communication, depending on the modulation, bit rate and center frequency (for more information see [11])

2.1.6 Mode F

Mode F is bidirectional, operating on $433MHz$ with chiprate $2.4kcps$ and NRZ encoding. Also recommended for long range communication. It is separated in two sub-modes which define the initiator of the communication. It can be either the metering device (similar to T2), or the data collector using a wake-up frame (like in R2).

2.2 Wireless M-Bus - Data Link Layer

EN 13757-4:2011 defines two different frame formats - A and B. In general, each data frame is composed of several data blocks with specific functions and definitions. The data is presented in hexadecimal numbers, where one byte of data consists 2 hex numbers.

2.2.1 Frame Format A

On fig. 2.2 is presented the construction of frame format A. It can be used by all the modes explained in the previous section.

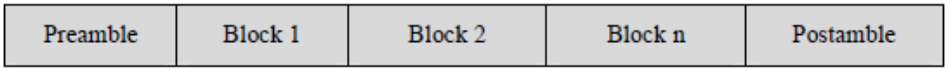

Figure 2.2: Frame Fromat A

It starts with **preamble**, used for synchronization between the transmitter and the receiver. It includes also the header of the packet. For each mode the size of the preamble is different.

- **Mode S:** The short preamble is 6 bytes and the long one is 72 bytes
- **Mode T:** 6 bytes
- **Mode R2:** 12 bytes
- **Mode C:** 8 bytes
- **Mode N:** depends on the used modulation (sub-modes respectively). For 4GFSK (sub-mode N2g) is 8 bytes, while for the rest is 4 bytes
- **Mode F:** 12 bytes

The next block is **Block 1** (fig. 2.3) This block is with fixed length and

L-field	C-field	M-field	A-field	CRC-field
1 byte	1 byte	2 bytes	6 bytes	2 bytes

Figure 2.3: Block 1

contains the following information:

- **L-field** - The length of the message that will be sent excluding the CRC bytes.
- **C-field** - Control field. Specifies the frame type. The content of this byte is presented on fig. 2.4. **RES**=0; **PRM** defines the source of the message (primary - 1 or secondary - 0 station); **FCB** is used for detecting frame duplication (alternates between 0 and 1 for successive frames from a primary to secondary station); **FCV** specifies if FCB is used; **ACD** if set to 1, indicates that the sending secondary station has high priority data available, which should be requested by the primary

station; **DFC** is the indicator if receiver's buffer is full and will not be able to process further frames. The last four bits define the **Function Code**. This determines the type of the frame being sent - SND-NR (Send, no response), SND-UD (Send a command), etc.

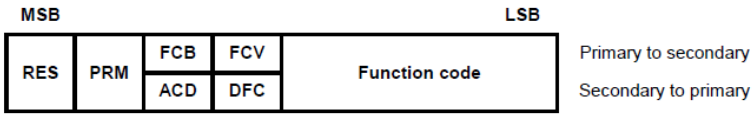

Figure 2.4: C-field byte content

- **M-field** - Specifies the manufacturer's ID according to ISO 646 - with three uppercase ASCII letters. There is a strict algorithm for creating it, so that each ASCII letter to be presented as a 5-bit long word, instead of 8-bit.

- **A-field** - Unique address for the device. If the MSB of the M-field is 1 - the address is "soft" (Programmed by the installer). Otherwise, the address is defined by the manufacturer. The field contains 4 bytes ID, 1 byte for device version and 1 byte for the type of the meter (water, electricity, gas, etc.). If the address is "soft" and all the meters have unique ID, then the remaining bytes can be used for specific purposes.

- **CRC-field** - a 2-byte Cyclic Redundancy Check for verification of the received data. It is calculated over the block where it is contained. The CRC polynomial is $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$ with initial value 0

Block 2, presented on fig.2.5, starts with a Control Information (CI) byte where is specified the type of the application protocol and the type of data that will follow in the Data-field.

CI-field	Data-field	CRC-field
1 byte	15 bytes or (((L – 9) mod 16) – 1) bytes	2 bytes

Figure 2.5: Blcok 2

In case the message is longer than 15 bytes, it is followed by additional n data blocks, shown of fig.2.6

Blocks 2 to n are optional and are based on the length of the message that is supposed to be sent. The **Postamble** block is a short bit sequence used by S, T and R2 modes.

Data-field	CRC-field
16 bytes or ((L – 9) mod 16) bytes	2 bytes

Figure 2.6: Block n

2.2.2 Frame Format B

Modes C and F optionally can use another frame format - B. The idea behind this format is to reduce the sent data. It is separated in 4 blocks, shown on fig. 2.7. The preamble is the same as for Frame Format A - 8 bytes for C-mode and 12 bytes for N-mode respectively. It is followed by Block 1, which contains the same information fields as A format, with this difference that it does not finish with CRC-field. The main difference between the two formats is the length of the data field that can be sent without being followed by CRC bytes. For B format, it is 115 bytes or $L-12$ bytes. Block 3 is optional and should be used if the length of the data field exceed the maximum allowed in block 2.

Preamble	Block 1	Block 2	Block 3

L-field	C-field	M-field	A-field
1 byte	1 byte	2 bytes	6 bytes

Block 2:

CI-field	Data-field	CRC-field
1 byte	115 bytes or (L – 12) bytes	2 bytes

Block 3:

Data-field	CRC-field
(L – 129) bytes	2 bytes

Figure 2.7: Frame Format B

It should be noted that the multi byte data is transmitted with LSB first (in both Data link and Application layers).

2.3 Wireless M-Bus - Application Link Layer

The Application layer is developed with the main purpose to be optimal with respect to devices' battery consumption, possible collisions and also processor regarded requirements like RAM and computational power. It starts with the Control Information (CI) byte, which distinguishes between the various telegram types, header length (if present) and application functions such as Error from device, Alarm from device, Transport layer, Network layer, Link Layer Extension, Response from device, Response from device with a compact frame, etc.

Based on CI, the next byte may contain data from the meter, or may have an extension of the link layer. This extension is used for two way communication and contains information about the synchronization, whether the frame is relayed or duplicated, priority indication. It can be short (2 bytes) and long (8 bytes), where the long one contains also information about the encryption of the frames. After the Extended Link Layer, another CI field follows.

In fig. 2.8 is presented the application layer with a long header (The long header can be observed as the transport layer).

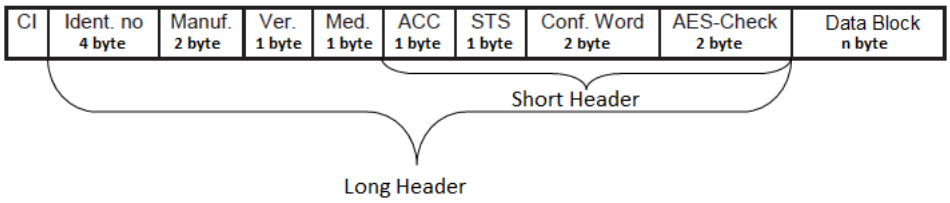

Figure 2.8: Application layer with long header

In the cases when the meter address is different then the address of the transmitting device, then a Long Header is expected, where the first 8 bytes contain the information about the metering device for its address, manufacturer, version and type (like in the Data Link Layer). The 4 bytes which follow afterwards, which are also part of the Short Header, are as follows:

ACC - Access number: For Wireless M-Bus devices this field is for clear indication of a new data telegram (for both, pushed and requested data). In case the telegram has the same ACC, the receiver should discard the telegram.

STS - Status byte: Depends on which is the transmitter (the meter or a supporting partner). If meter, contains information for the status of the device - low battery, application error, temporary error, permanent error (see fig. 2.9). If partner, contains information about the quality of the link between its receiver and the meter.

Conf. Word - Configuration word (also called configuration field) - contains information about the encryption and the number of encrypted bytes.

Bit	Meaning with bit set	Significance with bit not set
0,1	See below	See below
2	Power low	Not power low
3	Permanent error	No permanent error
4	Temporary error	No temporary error
5	Specific to manufacturer	Specific to manufacturer
6	Specific to manufacturer	Specific to manufacturer
7	Specific to manufacturer	Specific to manufacturer

Status bit 1 bit 0	Application status
0 0	No error
0 1	Application busy
1 0	Any application error
1 1	Abnormal condition / alarm

Figure 2.9: Status Field of the Meter

The presence of at least short header is recommended since it takes care of the security of the transmitted data. The last field - **AES-Check** is required for some of the encryption modes used by Wireless M-Bus (mode 5 and 6). It should be filled with data $2F_h$.

The header is followed by the a Data Block where is contained the real measurements information lead by service bytes used for defining the interpretation of the data.

2.3.1 Data block - Standard M-Bus frame

On fig. 2.10 is presented the structure of a regular Application Layer Data block.

DIF	DIFE	VIF	VIFE	Data
1 Byte	0 ... 10 (1 Byte each)	1 Byte	0 ... 10 (1 Byte each)	0 ... N Byte
Data Information Block DIB		Value Information Block VIB		
Data Record Header DRH				

Figure 2.10: Standard M-Bus Data Block

Every data record has a Data Record Header (DRH), which contains Data Information Block (DIB) with information about the length, type and

coding of the data (e.g. 4 digit BCD, minimum value), and Value Information Block (VIB) with information about the value of the unit and multiplier (e.g. $E0111nnn$ is Volume flow, l/h, with multiplier between 0.001 and 10 000 depending on the value of nnn, E is the bit for extension and is used if after VIF follows VIFE). The fields DIF and VIF are always present, while DIFE and VIFE are optional and not often used. One M-Bus frame can be (and usually is) consisted of concatenation of several data records.

2.3.2 Data field - Compact M-Bus frame

In most of the cases, the DRH fields from a metering device do not change for a very long period (or do not change at all). Therefore this information becomes redundant. In the same time, it may consist a very considerable amount of bytes. For example, if some of the transmitted values is not a metric unit, there is requirement of existence of a VIFE field with value $3D_h$, which informs that the value is in non-metric values and Table C.1 in [10] presents the units that can be used. In case, though, that value's unit does not exist also in this table, then there is option for sending ASCII string with the information of the unit string. Knowing that each symbol from the ASCII is presented with 8 bits, the value of VIFE can easily reach the limit of 10 bytes. If this value is the standard value the meter is supposed to transmit, the length of the frame is increased with 10 bytes only for additional information, that will never change. This makes the protocol inefficient. For this reason it is developed another M-Bus frame, called Compact.

The idea behind the Compact M-Bus frame is reducing service information data from the application layer by separating DIB and VIB from the M-Bus Application protocol data and adding two additional frame types to the standard full frame. It is required to be used in cases when the frame structure is unchanged for certain period. On fig. 2.11 are presented the structures of full, compact and format M-Bus frames. Operating with Compact M-Bus Frame may provide high improvement in the battery lifetime without losing important information.

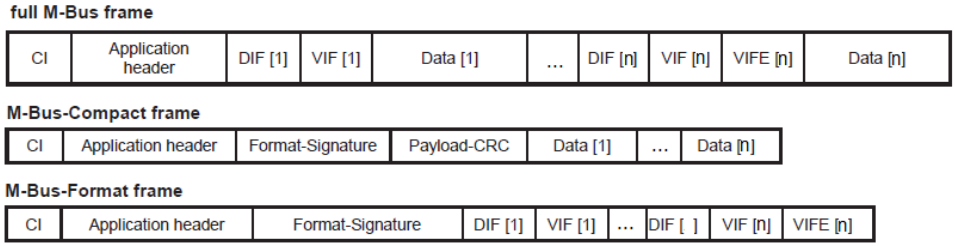

Figure 2.11: Structure of Standard and Compact M-Bus frames

When transmitting a compact frame, the Format-Signature is consisted of 2-byte CRC calculated over the M-Bus Format Frame starting from the first DIF until the last DIF/VIF. After that a 2-byte Payload-CRC, calculated over the data of the standard M-Bus frame starting from the first DIF until the last Data block. After the Payload-CRC follows a concatenation of all the data in the frame. It is recommended a full M-Bus frame to be transmitted after certain interval, even if the format of the frame does not change. Both fields - Format-Signature and Payload-CRC, are calculated over the polynomial, which is used for the other CRCs ($x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$ with initial value 0)

The Format Frame is used when the receiver requires information about the DRH. In this case it is transmitted either a full M-Bus and the information is gathered from there, or a Format Frame is transmitted.

2.3.3 Encryption

Wireless M-Bus has option for 2 types of encryption - *DES* and *AES-128* in different modes and starts right after the configuration word. In order to verify the correctness of the decryption, it should start with a known sequence. For the DES modes it is required that the data sequence should start with data point containing "date and time" (2 bytes). For AES it is a well known sequence - 2 bytes of $2F_h$.

Due to the mathematical nature of the encryption methods, the number of encrypted bytes should be multiple of 8 for DES and multiple of 16 for AES-128. In the cases where these requirements are not fulfilled, to the data has to be added padding "dummy" bytes until $data \pmod a = 0$ ($a = 8$ or 16, depending on the encryption).

It is also possible to have *partial* encryption. In these cases the encrypted data is followed by unencrypted one.

Chapter 3

Analysis

Once installed, the water meter becomes static. Other words, its location will not change in time and in most of its lifetime will transmit the same type of data. This gives certain flexibility for reducing the sent information and recover it based on indirect clues and a priori data taken from a database. Some examples for such a reduction are:

- Removing part of the identification number of the water meter;
- Giving a new, smaller, ID to the meter based on the area where it is located;
- Removing the DRH from the Data fields
- Removing some of the most significant bytes from the water consumption values. For example, if a water meter sends it's information once per minute, it is not necessary to send value for the current consumption 87654415 liters. It may be enough to send "415" and based on the ID and some data stored in a database, the collector will process the rest of the information.
- Sending information only for the change in the water consumption between the current and the previous transmission

The reading methods that are of particular interest for the scope of the project are "Drive-by" and "Fixed Network" due to the fact that currently they are the most commonly used methods for remote reading of water meters. So, the possible receivers can be:

1. in a vehicle for "Drive-by" data collecting;
2. a stand alone collector, located somewhere in the coverage of the transmitting meter, owned by the water-providing company (or other institution) used as part of a fixed network;

3. a stand alone collector, located in user's home for collecting the data from all the meters in his/hers home/office/etc;

4. a portable device, owned by the user of the water meter (for example USB stick for his/hers PC);

It should be considered that several different receivers (from the explained above) might be connected to one water meter. For example, the water providing company's reading method for certain area is "Drive-by" and in the same time the owners of one of the homes are equipped with smart meter data collector (fig. 3.1). This leads to the need the protocol to be such that gives the possibility all the receivers to be able to reproduce the data. For example:

- the meter sends message "I am alive, and there is no change/the change is X liters" - it will be suitable for fixed networks, but if the vehicle receive such information, it will not be possible to decode the value of the meter

- in a fixed network, established by the water provider company, may be required receiving of data $1 - 2$ times per day. On the other hand, if the owner has a "smart" home system and requires real time data from the water meter, getting the information so rarely is not acceptable.

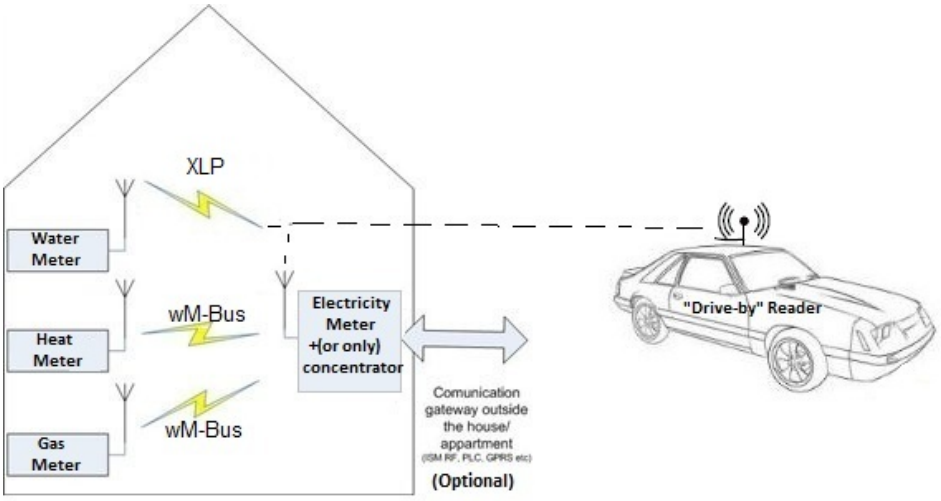

Figure 3.1: Scenario for use of XLP

As water meter, the device is not taking care of automation and control, but only of measuring. For this, it is acceptable if the meter has only one-way communication. The two way communication could be, if it is necessary, for

service (protocol updates, commands) or to shorten the transmission with the idea not to pollute the ether/to preserve energy (if the data is too much, it is useless to send the data if there is nobody, who is capable of receiving it).

Establishing of fixed network and implementation of AMI is promising and probably will be part of every city in the future. Though, time will pass before it is implemented everywhere, since it requires a lot of resources and regulations. Furthermore, not many homes are equipped with M-Bus infrastructures. For this reason one of the most frequently used methods for reading water meters is "Drive-by".

3.1 "Drive-by" meter reading

The general approach for "Drive-by" reading (which is also suggested in [11]) is using T- or C-mode where the transmission is very short and in a few seconds interval. Furthermore, C-mode has the option of using Frame Format B which has reduced CRC bytes and also, as mentioned before, is developed on the base of T-mode but with higher bit rate due to the different encoding. In addition, the frequent transmissions will allow a user with installed data collector (static or portable) to have almost real-time information about his/hers consumption.

Since the other parameters for these modes are the same, both can be received by the same receiver configuration (see below). After some discussions with Rasmus Jacobsen (see the Acknowledgements), it was concluded that currently, most of the companies who use Wireless M-Bus in their smart water meters which are suitable for "drive-by" reading, develop their applications on the base of C-mode. One example can be seen in [7] - a data sheet of a water meter with wM-Bus communication, operating in C-mode, which transmits data package every $16th$ second. In its entire lifetime this meter sends wM-Bus frames with the following pattern: 7 M-Bus Compact frames, 1 full M-Bus frame. Regarding the life time, we read the following:

- 12-year battery - 3.65 VDC, 2 A cells lithium
- 16-year battery - 3.65 VDC, 1 C cell lithium

3.2 Project Statement

The goal of the project is investigating ways for creating an eXtreme Low Power (XLP) protocol, based on reduction of Wireless M-Bus standard datagram. A requirement is that once the information reach the desired destination, to be possible to be recreated into a functional Wireless M-Bus message for further exploitation (for reference see fig. 1.6). For this reason it is examined the wM-Bus standard and the ways the data is created. Further

requirement is the same electronic equipment to be used also for the other $868MHz$ wM-Bus modes (see section 2.1). For this reason, the Physical layer will not be taken under consideration. Therefore, the investigations are focused on Data link and Application layers. Since the XLP concept is based on reducing redundant data, it will be investigated also the required type of data in the database necessary for recovering the wM-Bus datagram. Physical limitations in the receiver's side, like size, available memory, power constrains, etc. are not taken under consideration either. To ease the investigation, it will be assumed that only water meters operate with XLP. The protocol has to be suitable for "drive-by" reading. The water meter presented in [7] and its data are used as reference. The calculations for the radio module are based on Texas Instrument's CC1121 - High Performance Low Power RF Transceiver [14], operating on $868MHz$ with output power $10dBm$.

Chapter 4

Design Considerations

4.1 Cyclic Redundancy Check

The cyclic redundancy check (CRC) is a method for detecting errors that have occurred during the transmission. It is based on division of two polynomials in GF(2). It is implemented by adding redundant information, called *checksum* in the end of each transmitted frame. The content of these checksum bits is dependent on the message that is supposed to be transmitted and on a generating polynomial $G(x)$ (For wM-Bus $G(x) = x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$). The message can be observed as a big binary number $m_n m_{n-1} m_{n-2} \cdots m_2 m_1 m_0$.

The algorithm for creating CRC checksum is as follows:

1. The binary number's polynomial is $M(x) = m_n x^n + m_{n-1} x^{n-1} + m_{n-2} x^{n-2} + \cdots + m_2 x^2 + m_1 x^1 + m_0$.
2. Multiply $M(x)$ with x^k, where k is the degree of the generator polynomial $G(x)$.
3. Compute $x^k M(x) \pmod{G(x)} = R(x)$, where $R(x)$ is the remainder.
4. The message that will be transmitted in this case is $T(x) = x^k M(x) + R(x)$

On the other side, when the message $T(x)$ is received, the same operation with the same generating polynomial $G(x)$ is performed. If the result of the calculation $T(x) \pmod{G(x)} = 0$, then the message is assumed to be correct.

In case an error occur in the message, it can be presented as a polynomial $E(x)$ with different degree and terms, depending on the position and the number or alternated bits. In this case the received message is $T'(x) = T(x) + E(x)$. Since $T(x) \pmod{G(x)} = 0$, the remainder $R(x)$ can be observed as $R(x) = E(x) \pmod{G(x)}$. If $R(x) \neq 0$, then an error is detected

and the message is discarded. This means that the errors that have occurred during the transmission such that $E(x)$ is multiple of $G(x)$, will **not** be detected.

The Cyclic Redundancy Check is reliable method and if the generating polynomial $G(x)$ is chosen properly, it is capable of detecting errors like:

- if $G(x)$ has at least two terms ($G(x) \neq x^i, \forall i$), can detect every 1 bit, 2 bit and 2 adjacent bit errors
- if $G(x)$ is a multiple of $(x + 1)$, any odd number of errors
- burst errors with length equal or less then the degree of $G(x)$
- for burst error of length $k + 1$, the chance of not detecting an error is $(1/2)^{k-1}$

4.1.1 Number of undetected errors

Let's observe the sent message

$$T(x) = x^k M(x) + R(x), \tag{4.1}$$

where $M(x)$ is the actual message with degree i and $R(x)$ is the checksum with degree less than k. Then

$$\deg(T(x)) = \deg(x^k M(x)) = k + i, \tag{4.2}$$

and since $\deg(R(x)) < k + i$, the $\deg(T(x)) = k + i$. From this, the degree of the received message $T'(x)$ is also $k + i$, so $\deg(E(x)) \leq k + i$. We know that we have an error if $G(x)$ does not divide $E(x)$. But in the cases where $G(x)$ divide $E(x)$, we are not able to detect the error.

Let's assume that $G(x)|E(x)$, then there exist a polynomial $Q(x)$ such that

$$E(x) = Q(x)G(x) \tag{4.3}$$

The degree of $Q(x)$ is then given by

$$\deg(Q(x)) = \deg(E(x)) - \deg(G(x)) \leq k + i - k = i. \tag{4.4}$$

So there exist $Q(x) = 2^{i+1}$ different polynomials which will give the result $E(x) \pmod{G(x)} = 0$ and from this, $2^i + 1$ different messages $T'(x)$ which give $T'(x) \pmod{G(x)} = 0$.

Considering that the length of the message $T(x) = i + 1 + k$ bits, which gives 2^{i+1+k} different messages, then it can be denoted that a CRC with a polynomial with degree k can detect

$$1 - \frac{2^{i+1}}{2^{i+1+k} - 1} = 1 - \frac{1}{2^k - 1}. \tag{4.5}$$

of the error cases.

With respect to the wM-Bus polynomial, this is $(1 - \frac{1}{2^{16}-1} = 99.998\%$ of the errors.

4.1.2 Format Signature and CRC

As discussed in section 2.3.2, since the information in the Data Record Header (DRH) is considered as redundant, a Format Signature is computed over all DRHs in in the same way the CRC is calculated and with the same generating polynomial $G(x)$. So, the compact frame contains this CRC in the form of Format Signature instead of all DRHs. But applying this approach, as was shown previously in this section, there is certain chance that different combinations in the DRH may give the same Format Signature, which will lead to a problem regarding the decoding of the received message from the receiver.

If the concatenation of all the DRHs in one wM-Bus frame is observed as a message $M(x)$ with degree i, and a Format Signature in the form of CRC is calculated over it. Let's denote a presence of error $E(x)$, which is actually the presence of a DRH with other values. In this case the $\deg(E(x)) \leq \deg(M(x)) \leq i$. Therefore the coefficient polynomial $Q((x))$, based on equation 4.4 can be denoted that

$$\deg(Q(x)) = \deg(E(x)) - deg(G(x)) = i - k, \tag{4.6}$$

therefore 2^{i-k+1} different possible options for DRH, which will provide the same Frame Signature. For example, if the concatenation of all DRHs in a wM-Bus frame is $B = 10$ bytes (80 bits => $\deg(B(x)) = i = 79$), knowing that the FS is 2 bytes (16 bits), calculated with generating polynomial $G(x)$, where $\deg G(x) = k = 16$, it leads that depending on the values in the DRHs, the number of words that will have the same FS is $B = 2^{79-16+1} = 2^{64}$. Despite the fact that the number of combinations that will give the same Format Signature (FS) is $\frac{2^{i-k+1}}{2^{i+1}} = \frac{1}{2^k} = \frac{1}{2^16} = 0.0015\%$ of all the possible combinations, this is quite large number and that might lead to complications when messages with different data structures calculate and send the same FS.

A CRC with the right polynomial is very good for detecting errors which can occur in a communication channel during the transmission, due to bursts or electrical spikes. In the cases, though, where the data can be completely different from the beginning to the end, is not that efficient.

4.2 Number of meters in receiver's range

The number of meters in the receiver's range depend on number of parameters such as:

- Transmitter's antenna, output power and position
- Receiver's antenna and its location
- Presence of obstacles between the communicating devices
- The size and nature of these obstacles (concrete wall, metal wall, trees, etc.)

4.2.1 Coverage area

According to Friis transmission equation, the input power of the receiving antenna P_{RX} in terms of decibels (dBm) is

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} - Loss - 20\log_{10}(\frac{4\pi d}{\lambda}) \tag{4.7}$$

where

P_{RX} - Input power of the receiving antenna

P_{TX} - Output power to the transmitting antenna

G_{TX} - Gain of the transmitter antenna

G_{RX} - Gain of the receiver antenna

$Loss$ - Losses (due to antenna mismatches, polarization, reflection, fading, environmental conditions, etc.)

λ - wavelength of the carrier signal ($\lambda = \frac{c}{F_c}$, where c is the speed of light and F_c is the carrier frequency)

d - distance between the two communicating devices

If ideal conditions are assumed ($Loss = 0$), for $P_{TX} = 10dBm$ and $P_{RX} = -100dBm$ and carrier frequency $f_c = 868MHz$ (the sensitivity of the receiver), the reading distance can reach almost $9km$. In reality, though, considering the locations of the two devices (Receiver's antenna not higher than $2m$ above the ground level and Transmitter located in a basement/closet/meter pit), the antenna mismatches and return losses, and the environment (many obstacles), the losses can easily reach at least $Loss = 20 - 25dB$ (see fig. 4.1). On fig. 4.1 is presented the received power as a function of the distance between the two communicating devices for different losses. As can be seen, a small deviation in the losses may lead to significant loss of coverage distance.

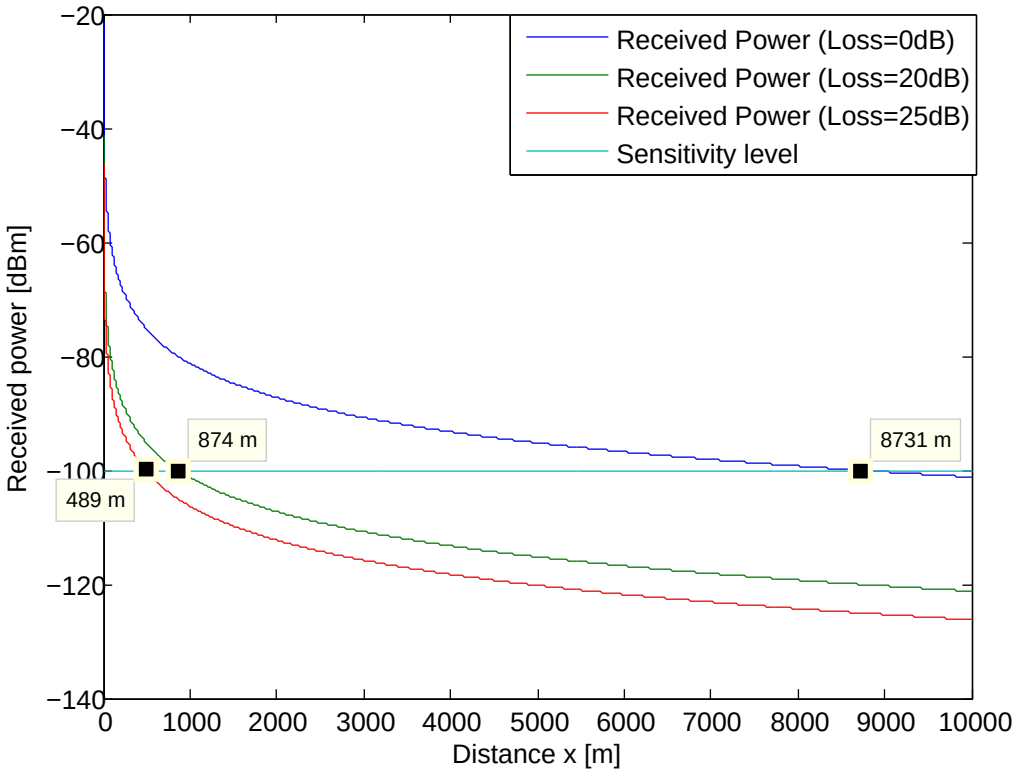

Figure 4.1: Received power as a function of the distance between two communicatin devices

4.2.2 Water meter density

On table 4.1 is presented the population density per km^2 for some of the most populated cities in Europe. According to this source, the most dense populated city in the world is Mumbai with $29650 ppl/km^2$.

It should be considered that in some areas of those cities the population density is much higher than in other areas. Also should be noted that usually for several people is counted one meter (since in one home usually live 2,3 and more people, using one water meter). For this reason an assumption of ≈ 10000 water meters per km^2 is considered as relatively acceptable.

Assuming the range of the receiver is $d \approx 560m$, then the coverage area is $S = \pi d^2 \approx 1km^2$, therefore in the coverage of the receiver can be expected up to 10000 water meters. This value is highly overrated, due to the fact that this coverage range could be reached in rural areas, where not many obstacles are between the transmitter and the receiver. The higher the density of meters is, the more obstacles are in the area (e.g. buildings). Considering that the high frequencies have very low wall penetration, the communication range will be reduced rapidly.

country	population density (ppl/km^2)
St. Petersburg	8550
Istanbul	7700
Athens	5400
Madrid	5200
London	5100
Munich	3100
Stockholm	2700
Rotterdam	2500
Copenhagen	1850

Table 4.1: Population density, people per square kilometer, for some of the most populated cities in Europe (source: http://www.citymayors.com/)

4.3 Data from the meter

The reference water meter used in this project - Kamstrup's Multical 21 [7] transmits the following information:

- Current meter reading
- Meter reading on the first day of this month. Alternatively, max. flow during latest completed month
- Operational Hour counter
- List of active info codes
- List of info codes which have been active within the latest 30 days (and how long they have been active).

On fig. 4.2 are presented device's display and list of info codes. As can be seen, the information about the water consumption is presented with 8 digits and shows information down to liters. The indicators under the digits are the info codes. With respect to the transmission, these codes are presented as a bitmask (active/no active). Due to the fact that further information was not provided regarding these info codes, it is assumed that the meter transmits one byte for the currently active codes and one byte for the ones active in the past 30 days. The information about the time the info codes have been active, according to [7], is hourly rounded. This means that the value is maximum $31 \cdot 24 = 744$ hours.

Since the exact information that Miitor's meters will transmit is not defined yet, for the purpose of this project it will be assumed they transmit the same information.

Info code flashes in the display	Meaning
LEAK	The water in the meter has not been stagnant for one continuous hour during the latest 24 hours. This can be a sign of a leaky faucet or toilet cistern.
BURST	The water consumption has been consistently high for half an hour, which indicates a pipe burst.
TAMPER	Attempt of fraud. The meter is no longer valid for billing.
DRY	The meter is not water-filled. In this case nothing will be measured.
REVERSE	The water flows through the meter in the wrong direction.
RADIO OFF	The meter is still in transport mode with the built-in radio transmitter turned off. The transmitter turns on automatically when the first litre of water has run through the meter.
■■ (two squared "dots")	Two small squares flashing alternately indicate that the meter is active.
'A' followed by a number	Indicates that the meter has been checked and given a new revision number.

Figure 4.2: Display and Info codes for Multical 21 [7]

4.3.1 BCD or Binary

Whether the meters transmit their values in BCD (Binary Coded Decimal) or binary format, it is defined in the DIF field of every data block. Taking again for example the meter from [7], its value is from 00000000_{dec} to 99999999_{dec} (10^8) liters. If decoded as BCD, the value should be sent with $8 \cdot 4 = 32$ bits, where every digit is presented with a 4-bit binary number. In the same time, $99999999_{dec} = 10111110101111000011111111_{bin}$, which is a 27-bit binary number. Though, with respect to the transmission, the value has to be transmitted as a byte sequence with LSB first, so the value will be presented as a 4-byte (32-bit) number, where the 5 MSBs are 0 . In this case, with respect to the transmission time, there is no difference if the value is BCD or binary presented.

Regarding the time the codes have been active, the information should be transmitted with 2 bytes and again does not matter whether it is in BCD or binary format. Though, since the value cannot exceed 744, an option is to "merge" values, in case that more codes have been active. Since it is required max 12 bits for each value, 2 timing values could be written in 3 bytes.

A concept for saving power by reducing the transmission time leads to idea not to transmit information, contained in the most significant digits of the transmitted data. For this reason, let's observe the value field. If removing the two MSBytes, the meter will transmit information for up to 9999 liters if BCD and up to 65 535 liters if binary. This is 6.5 times more information within the same transmission time. For this reason, it is better at least the values for the consumption (historic and current) to be transmitted in binary format.

4.3.2 Water consumption

On fig. 4.3 is presented the annual water consumption per person for year 2000 in several countries. As can be seen, USA are leaders in this chart with $1682m^3$ per capita. This is about $100m^3$ per month. It has to be noted that this chart considers also the water used by the industry. According to this source of the chart (www.conferenceboard.ca), the people in Canada in their home use about 300 liters per day. Therefore, for an average house hold, the monthly increase of the water meter value can be approximated to $45m^3$ (assuming 5-member family). In the same time the industry uses "68 per cent of the total water used in Canada". This makes $85m^3$ per capita per month in Canada.

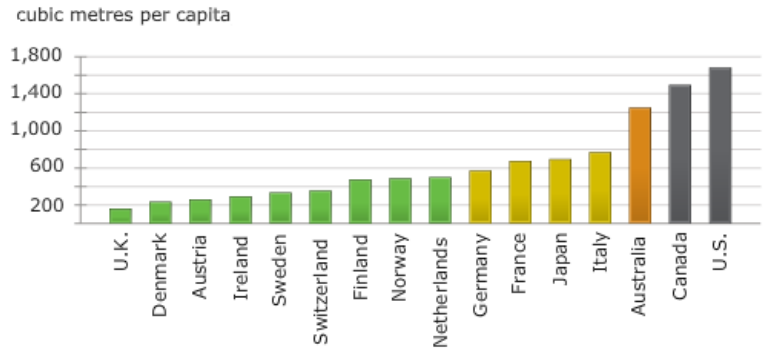

Figure 4.3: Annual Water Consumption per capita $[m^3]$ (source: http://www.conferenceboard.ca/)

Taking under consideration all these values, it is assumed that for one month the water consumption detected by a single water meter will not exceed $1000m^3$ and this value will be considered as a limit for this project. This will allow the results from this project to be applicable for water meter used in house holds and industries with monthly water consumption up to this level. In cases where this consumption is higher, additional investigation should be provided. Another investigated limit will be $100m^3$ for domestic water meters.

Chapter 5

XLP

The structure of the XLP protocol is meant to save energy on the meter side. Since in most of the cases the meter transmits a regular message with data for its values, it is of interest the XLP to take over when such message is about to be transmitted and reduce it as much as possible. Limited operation options will provide the possibility of reduction of the transmitted package even deeper by removing some of the control bytes that are necessary for proper interpretation of the received data. The principle of creating an XLP should be in a sense the same as the creating of Compact M-Bus Frame. For this reason a regular wM-Bus message has to be prepared from the microprocessor unit of the meter device. The transition **wM-Bus-to-XLP** should be implemented on top. This will provide more flexibility of the devices (despite the fact that the MCU has to perform several more operations).

For a wM-Bus, the content of the frame is interpreted by reading the $C-field$ from the Data Link Layer and the $CI-field$ from the Application Layer. Since the idea is the XLP to be active only under standard conditions - broadcasting of normal values with no priority, it should activate only if the content of C-field is $44h$ (SND-NR, message from a primary source) and the content of the CI-field is "Response from a device" - $78h$, $79h$, $69h$, $7Ah$, $7Bh$, $6Ah$, 72, 73 or $6B$ (full, compact and format frame formats with no-, short- and long application headers respectively). In cases where an extraordinary message, such as alert, is supposed to be transmitted or a 2-way communication is initiated, the transition **wM-Bus-to-XLP** should not be established.

5.1 Preamble and mode distinguishing

Whether it is T-mode or C-mode, the receiver distinguishes when the preamble (which also contains the synchword) is received. In table 5.1 are presented the preamble blocks for the two modes. As can be seen, C-mode's

first 6 bytes are the same as T-mode's and after that is followed by two additional bytes which define the C-mode and the frame used format (C(A) or C(B)). The receiver can clearly recognize that this is C-mode synchword, but not data from T-mode since in the specified wireless M-Bus' $3-out-of-6$ encoding there is no combination 010101 (see Appendix A). The last 8 bits define the frame format (11001101 for format A and 00111101 for format B)

mode	preamble chip sequence	encoding
T1	19*(01) 0000111101	3-out-of-6
C1(A)	19*(01) 0000111101 01010100 11001101	NRZ
C1(B)	19*(01) 0000111101 01010100 00111101	NRZ

Table 5.1: Preambles for T- and C-modes

The $3-to-6$ encoding (or $3-out-of-6$) is a modification of $m-of-n$ code, where $n-$bit word contains m instances of a "one". For the $3-out-of-6$ code the transmitted words are $6-$bit consisting 3 1's and 3 0's. This allows

$$\binom{6}{3} = \frac{6!}{3!(6-3)!} = 20 \tag{5.1}$$

different combinations. Considering that the wireless M-Bus protocol information is presented as hexadecimal numbers, this encoding uses 16 of the 20 different bit sequences. In appendix A is presented a table with the 3-out-of-6 encoded numbers in T-mode. Starting from 0_{dec} (0000_{bin}) and finishing with 15_{dec} ($1111_{bin} = F_h$), all the 16 numbers are presented with 6 bits. This leads to four sequences that are available and not used in the T-mode. One of them (010101) is used for defining the C-mode. The other three are free. It can be easily implemented one of the residual three combinations (for example 101010), followed by 2 additional bits, to be used for defining the XLP protocol. Since the idea behind the protocol is to be as short as possible, different data formats will not be used, which leads to the possibility of removing the final $8th$ byte of the C-mode preamble. In this case the preamble for XLP will be defined as:

19*(01) 0000111101 **101010XX**, where the first 6 bytes are like T-mode's, and the $7th$ byte defines that the information is in XLP format.

With such a created preamble, it gives the opportunity of having the same radio equipment for different implementations and flexible solutions with switching between the different modes.

5.2 XLP and Data reduction

The data reduction can be established in both, Data link and Application layers. In this section is given short overview and observation about these possibilities.

5.2.1 Frame Format

Wireless M-Bus suggests two frame formats - A and B. Depending on which one is used a number of CRC bytes can be reduced. Though, with increasing the amount of data covered by a single CRC check, the chance for undetected errors rises.

5.2.2 C-field

Since the previously defined standard conditions are based on C-field = $44h$, and the presence of XLP is already recognized in the preamble, this byte can be removed from the XLP datagram.

5.2.3 Meter ID

The meter ID is 8 bytes, where 2 bytes are manufacturer, 4 bytes are address, 1 byte is version and 1 byte is type. According to [10], if MSB of the manufacturer's field is 1, then the A-field can be "soft" address and as long as there is unique address for each meter in the coverage area, the rest of the bits can be used for user specific purposes. According to section 4.2.2, the water meters in an area covered by a receiver are no more than 10 000. For this reason if it is transmitted only 2 bytes of information, it will be sufficient to provide unique ID for every meter in the coverage area of the receiver ($2^{16} = 65536$ unique IDs). On the other hand, if 2 or more meters have the same ID and all the meters are in the receiver's database, based on different criteria, it can be denoted which one is the transmitting one. This may provide the opportunity of transmitting only 1 byte of the Address field. Such criteria can be:

- the current location of the vehicle (for "drive-by" readout),
- the RSSI from the receiver
- the data content of the meter. (e.g. if the water meter sends information about its current value, it should be incremental, so based on the last record, if the data is with lower value, then this is not the meter that is currently read)
- after implementing a CRC-payload check the values should match.

5.2.4 CI-field

The CI field, as mentioned before, shows the type of the data that will follow after. If the XLP is only for one strict data type, e.g. "M-Bus Response from device with short header" ($7A$), then the CI field also can be excluded from the datagram of the XLP.

5.2.5 Data Header

- Long Data Header - Considering that the meters will be with build-in transmitter, the long header will be discarded and not investigated.

- Short Data Header - The Status and Access Number bytes are not of particular importance and can be skipped and $wM-Bus-to-XLP$ transition to be applied when the value of the Status Field is 00_h (no errors), but the configuration bytes take care of the encryption and for wireless transmission it is mandatory to have certain protection. Though, the incrementation of the access number might be used for detecting malicious intervention. Since on every next transmission it should be incremented with a specific number, if an extra radio device is used to send the same data all the time in order to hide the value of the meter, it can be detected. The Access Number is also used as part of the initialization vector for one of the AES-128 encryption modes. Furthermore, in the status byte there is a lot of information that can be found helpful, plus there are 3 specific to manufacturer bits that can be used to support the XLP datagram.

- No Data Header - The shortest possible option. Though, this means that there is no information about the Access Number, Status and worst of all, no encryption of the data.

5.2.6 Data fields

With respect to reducing the redundant data in the telegram, the Compact M-Bus frame presents very optimal solution. All the Data Record Headers (DRH) are removed and instead of transmitting them, a calculated format signature is transmitted. When received, the receiver checks in its database if the received Format Signature (FS) is available and makes the decoding of the data based on it. In this case, the second CRC (Payload-CRC) is also required, since it is calculated over the whole M-Bus data block and gives information about the correctness of the assumed data. On the other hand, depending on the type of blocks "Data 1 $\cdots$ Data n" some of the most significant bytes (MSByte) may be possible to be reduced if they do not change often. The reproduction of the reduced data could be accomplished with the help of proper database with historical values for this data and the CRC-payload.

5.2.7 Format Signature

As it was discussed in section 4.1.2, an automatically CRC generated Format Signature may lead to some complications if two different DRHs give the same CRC result. Furthermore, it 2 bytes. On the other hand, in the Status

byte of the header (STS), there are 3 bits for manufacturer specific use. One of them (e.g. $STS[5]$) can be used as a flag for "default" Data Field, which can allow the reduction of FS. In case that one meter transmits XLP frames with different types of data, it could be implemented a table with all the possible data structures that it may possibly transmit. Such, 1 byte will give the opportunity of having 256 different data structures, which are enough.

5.2.8 XLP - final look

Based on the previous discussions in this section, a proposal for an XLP frame content is:

- 1 byte length information
- 2 bytes address
- 6 bytes short header (including two $2F_h$ bytes for AES-128 or two bytes with date information for DES encryption)
- 1 Format Signature word (optional)
- 2 bytes CRC-Payload
- n bytes data

For standard conditions are considered $C = 44_h$ and $CI = 7A_h$. The value of n depends on the amount of data that has to be sent. Refering to the meter from [7], as it was discussed in 4.3, the data that is supposed to be transmitted most of the times will be 4 bytes for current value and 4 bytes for a historical value (if no reduction is implemented there). Including also the info codes (which are not present always), the number of bytes may reach maximum 20 bytes. In this case, the worst case scenario for a XLP frame transmission is 32 bytes. For this reason, Frame Format B is considered as reasonable solution.

Such encoded, the XLP frame is not compatible with the standard wM-Bus receivers. Though, due to the nature of the frame, receivers created for XLP will be able to read also standard wM-Bus messages. For the sake of compatibility and to ensure the possibility the reader to "learn" quick, it is preferable if the meters send a regular wM-Bus message once in a while (as is suggested also for the Compact M-Bus frame).

On fig. 5.1 is presented the algorithm for deciding whether XLP frame to be created or not. A regular wM-Bus message is created and in case that the requirements for standard message are fulfilled ($C = 44_h$ and $CI = 7A_h$), and a standard wM-Bus message has been transmitted soon ($Count < 7$), an XLP will be created.

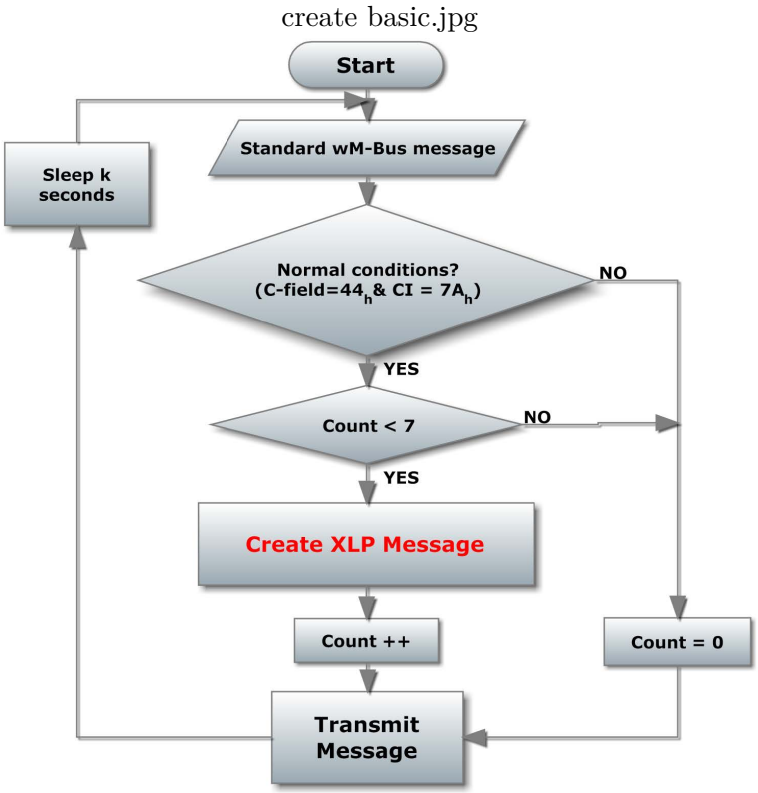

Figure 5.1: Algoritm for deciding wheter to create XLP frame

5.3 XLP message encoding

On figure 5.2 is presented algorithm for creating (encoding) a XLP message. It starts after the standard conditions are true (see fig. 5.1). In case that the Data Record Headers for all types of data in the Data Field are the default ones, flag is raised ($STS[5] = 1$, STS is the Status byte from the Header). Otherwise, a Format Signature is created and $STS[5] = 0$. CRC-payload is calculated over the Data Field of the regular wM-Bus message and DRHs are removed. In case that some of the data is supposed to be reduced, the corresponding bytes are removed. Then the new Data Field is a concatenation of the CRC-payload, Format signature (if such), and the Data values. The Data Field is attached to the rest of the fields, specified in 5.2.8 (Length, ID, header). Then the length of the message is calculated (including the 2 bytes CRC and the "dummy" bytes used for fulfilling the encryption requirements) and the result is put in the L-field; a CRC is calculated over the whole frame and is placed at the end.

5.4 XLP message decoding

The algorithm for decoding an XLP is separated in 2 parts - identifying the XLP (fig. 5.3) and the actual decoding (fig. 5.4). If the received message have the XLP preamble, and the meter is properly identified, the message is decrypted and the decoding of the XLP starts.

During the decoding (fig. 5.4), the first step is to recover the missing bytes from the Data Link layer. The C-field is filled with 44_h, the 8 identification bytes are filled with the original ID of the meter (taken from the database), $CI = 7A_h$, the Header and the Initialization Vector ($2F_h$ or date) are the same, so they are placed directly. In case $SYS[5] = 0$ and the Format signature is not present in the database, the message is discarded. Otherwise, a new, empty array, is created with the size of the regular wM-Bus data field that is supposed to be recovered (this information is also taken from the database). Then, fill the DRH bytes with the corresponding values, fill the bytes for which is available information from the XLP message and the rest of the bytes fill with the data stored in the database. A CRC is calculated over the restored Data Field and is compared with the CRC-payload, (which is part of the XLP message). In case they don't match, the bytes, taken from the database are incremented one by one until the two CRCs match. In this moment the message is considered as decoded and all the necessary data for creating a standard wM-Bus message are available. Then this data is processed according to the specification of the receiver.

create2.jpg

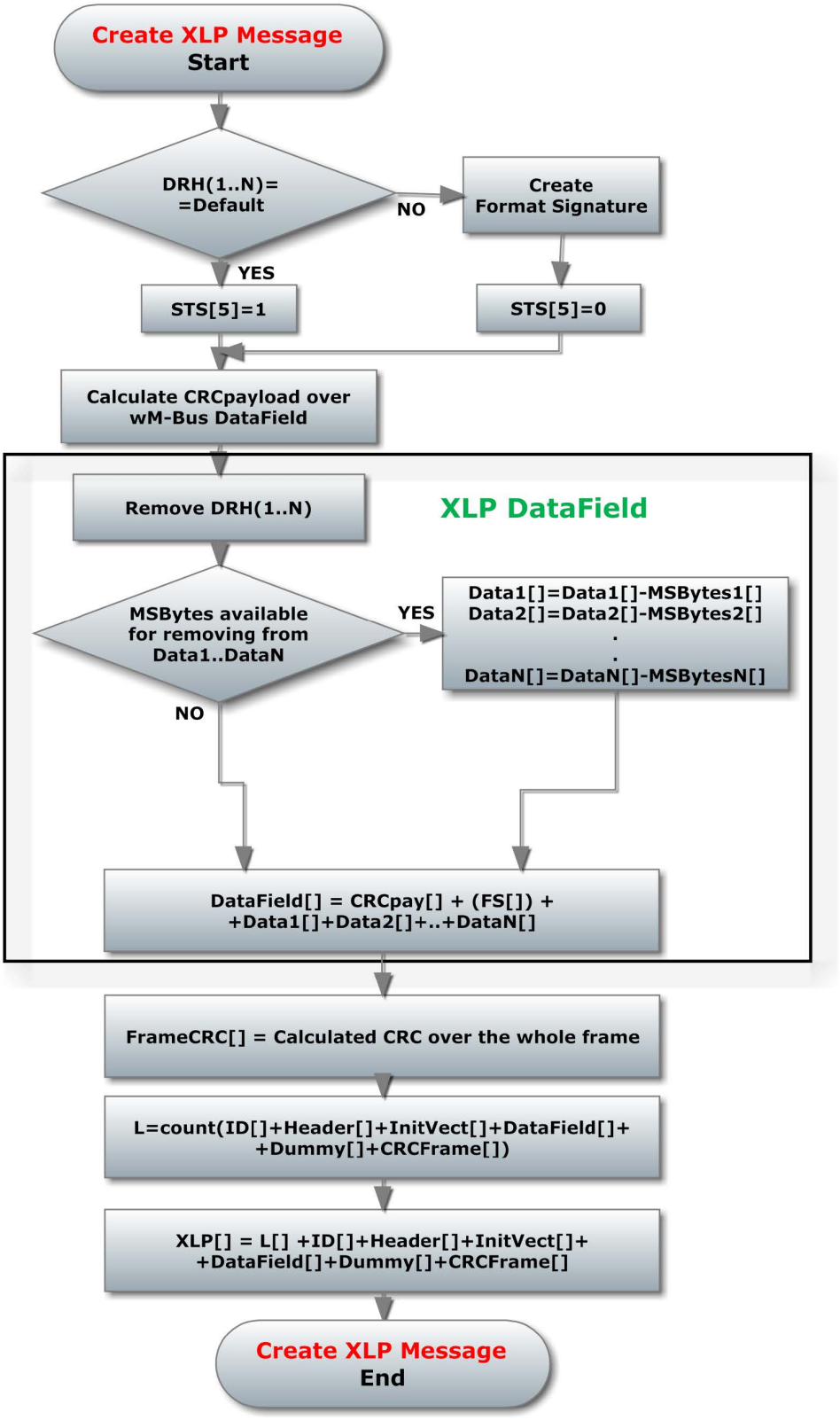

Figure 5.2: Algorithm for creating XLP message

Decode basic.jpg

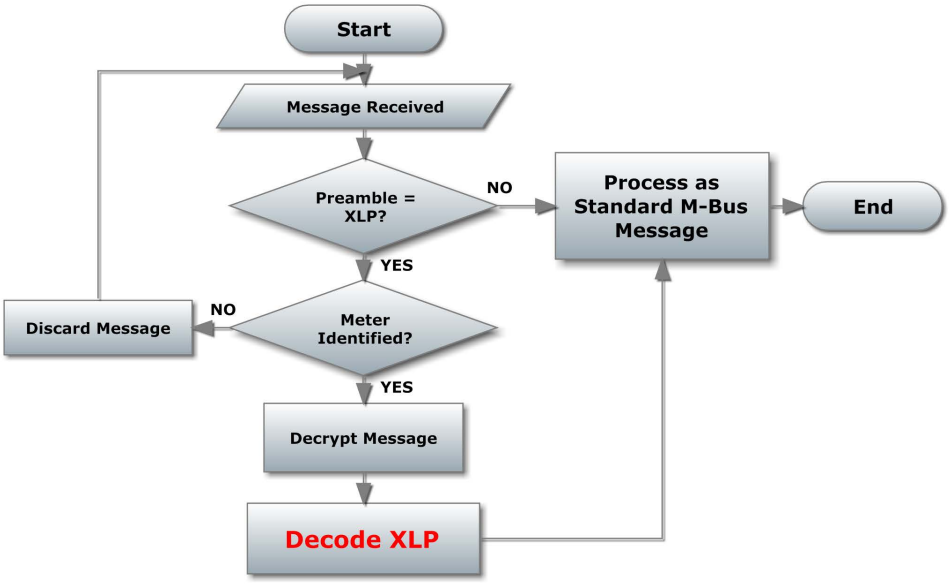

Figure 5.3: Identifying the XLP message

decode.jpg

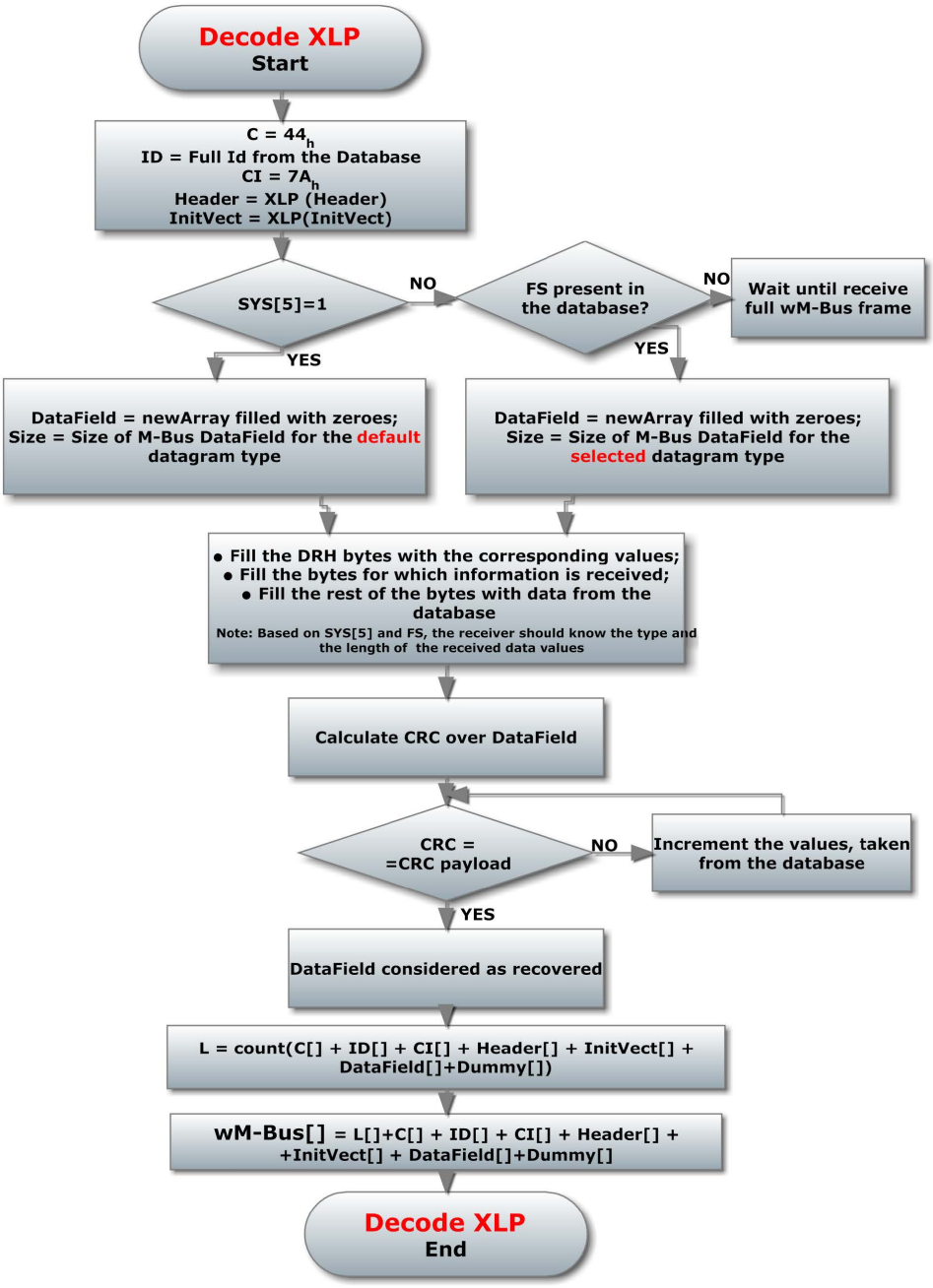

Figure 5.4: Decoding the XLP message

Chapter 6

Tests and results

It was performed simulation test in order to investigate the possible reduction of the Data field bytes. It is assumed that water meter from some "dummy" manufacturer and with "dummy" address transmits information for "normal" conditions ($C-field = 44_h$, $CI = 7A_h$), the info codes have 0_h values (so, information about the active hours is not transmitted), and operating hours are not transmitted. It is assumed that the meter is correctly identified, the information is properly decrypted and there are no errors due to noise in the communication channel (message is received correct). The other assumption is that the transmitted values are:

- $Data1$ - Current Value of the meter
- $Data2$ - Value for the first measurement of the month

and are presented as 4-byte binary numbers.

The DIF and VIF values are assumed to be as follows:

- Current Value: $DIF1 = 04_h$ (32-bit binary number), $VIF1 = 13_h$ (Volume in liters down to 1 liter)
- Value for the first time of the month: $DIF2 = 44_h$ (32-bit binary number, historical value), $VIF2 = 13_h$ (Volume in liters down to 1 liter).

Simulations for different transmissions are made, but the change of these values are not taken under consideration due to the reason that they are not object of investigation. It should be noted that their size will not change for the different data types used in the simulations.

In the simulations is not taken under consideration also the case when the value shown by the water meter reach its maximum value and start from 0 again.

It is **important to be noted** that the algorithm is created based on values calculated in a time line as follows: Database record - $Data2$ - $Data1$ (see fig. 6.1) and will not work for other cases!

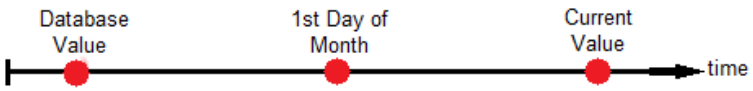

Figure 6.1: Time line of the measured values

In the simulations are considered 4 major different cases. As it was discussed in section 4.3.2, the average monthly water consumption in a regular household is $45m^3$, and it was assumed that the value of $1000m^3$ is sufficient also for industry purposes. Regarding the data that should be transmitted from the water meter, the traditional water meters measure down to $1m^3$ and the one observed in this project [7] gives information down to 1 liter. For this reason, the observed cases are:

- Maximum water consumption - $1000m^3$, resolution - 1 liter
- Maximum water consumption - $100m^3$, resolution - 1 liter
- Maximum water consumption - $1000m^3$, resolution - $1m^3$
- Maximum water consumption - $100m^3$, resolution - $1m^3$

For each case is made observation for the number of properly reproduced data values for different reductions of $Data1$ and $Data2$.

In the end is made a comparison between a standard wM-Bus frame, Compact frame and XLP frame for the same "default" message.

6.1 Architecture of the simulation engine

On fig. 6.2 is presented the architecture of the simulation engine used in this work. For its creation was used the programming language $C\#$.

In object $Meter$ is created the meter data using the following steps:

1. It is created a $DataField$, in the form of M-Bus message, with random value between 0 and 97 999 999 (volume in liters) or 97 999 (volume in m^3). This is used to simulate a value from the last reading in the Database of receiver Rx.

2. The value $Data2 = DataField + Random$ - random incrementation with value between 0 and 1 000 000 for liters (or 1000 for m^3)

3. $Data1 = Data2 + Random$ - another random incrementation in the same range as the incrementation of $Data2$.

4. A standard Data Field of a M-Bus message is created with structure shown on fig. 6.3

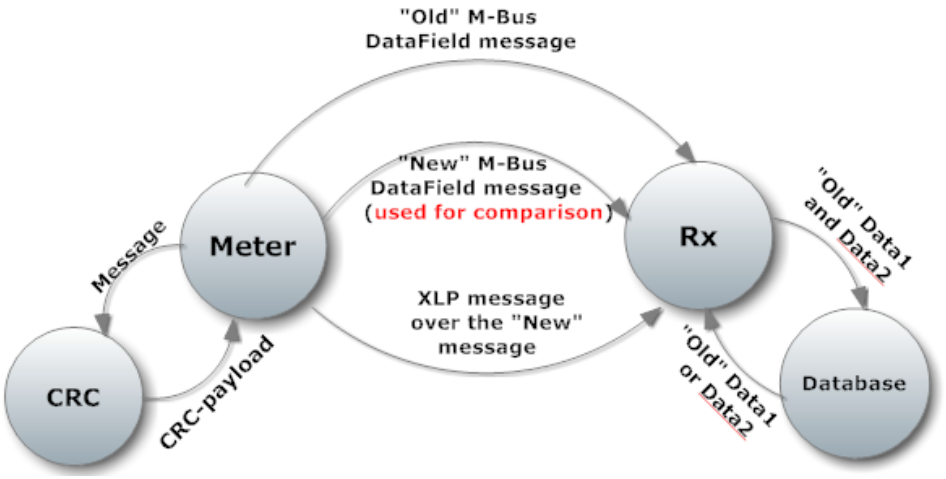

Figure 6.2: Architecture of the simulation setting

5. A CRC is calculated over this message in object CRC

6. An encoded (XLP) message is created by removing some of the most significant bytes (MSBytes) from the standard message and a CRC payload is attached to the beginning of the message (fig. 6.3)

Standard Message

DIF1=04_h	VIF1=13_h	Current Value	DIF2=44_h	VIF2=13_h	1st Day Value
1 byte	1 byte	3-4 byte	1 byte	1 byte	3-4 byte

Encoded Message (XLP)

CRC payload	Current value	1st Day Value
2 byte	1-3 byte	1-3 byte

Figure 6.3: Standard and XLP messages created in the simulation

The meter sends both Standard and XLP messages to the receiver Rx. In the receiver the XLP is decoded with the following algorithm:

- An empty array with 12 bytes is created
- In bytes $1, 2, 6$ and 7 are placed the DIF and VIF values
- The data from the transmitted XLP message is placed in the corresponding fields[1] and the rest of the data is taken from the database (see fig. 6.4).

[1]Note that the receiver should be aware with the number of removed bytes from each data (in this case $Data1$ and $Data2$)

- After the frame is filled with data, a CRC is calculated and is compared with the CRC-payload, which is sent with the XLP.
  - If they are equal, the message is considered as decoded
  - If not, it is performed incrementation of the bytes taken from the database. After each incrementation a new CRC is calculated and compared with the CRC-payload until a match occur.

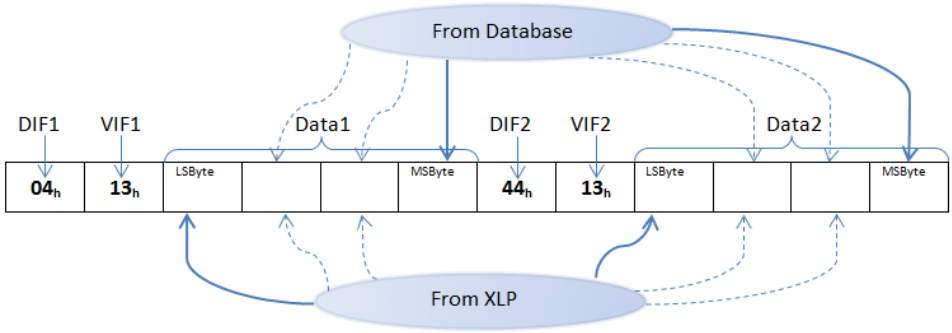

Figure 6.4: Decoding the XLP message

The incrementation of the bytes, taken from the data base is different, depending on the reduced from the original message bytes. It considers two cases of data reduction - equally removed amount of bytes from each data ($Data1$ and $Data2$), and case when from $Data2$ is removed 1 byte more than from $Data1$. The incrementation is implemented in a way to save computing power and improve accuracy. It is based on the fact that $Data2$ can never exceed $Data1$.

- when equal bytes are removed: increment $Data1$ with i = from 0 to $1FFFFF$ with step 1 and for each incrementation, check all the possible options for incrementation of $Data2$ from 0 to i. The value $1FFFFF$ is chosen high enough, so the incrementation will reach the limit of the meter value. In reality, this value is never reached, since matching CRC is found much before $i = 1FFFFF$.

- when $Data2$ has 1 transmitted byte less than $Data1$: in this case $Data2$ has one LSByte more taken from the database, so incrementation of this byte has to be performed, too. For this reason, in $Data2$ for each i, increment $Data2$ from 0 to $255 + (i * 256)$, and after each incrementation is checked if $Data2 > Data1$.

After the message is considered as decoded, it is compared with the Standard message that was transmitted in the same time with the XLP. If they are not equal, a counter for the errors increments its value.

The whole procedure is repeated number of times (between 1 000 000 and 1000), depending on the amount of calculations that have to be performed. The reason some of the tests to be repeated less times then others is based on the lack of computing power. Furthermore, the big amount of repetitions was established in the cases when there were no errors detected, or they were very small amount. In the cases where the errors exceeded $10 - 15\%$, the data reduction was considered as unappropriate and more then $100 - 1000$ repetitions were considered as unnecesary. On fig. 6.5 is presented a screenshot of the GUI implemented for the simulation engine.

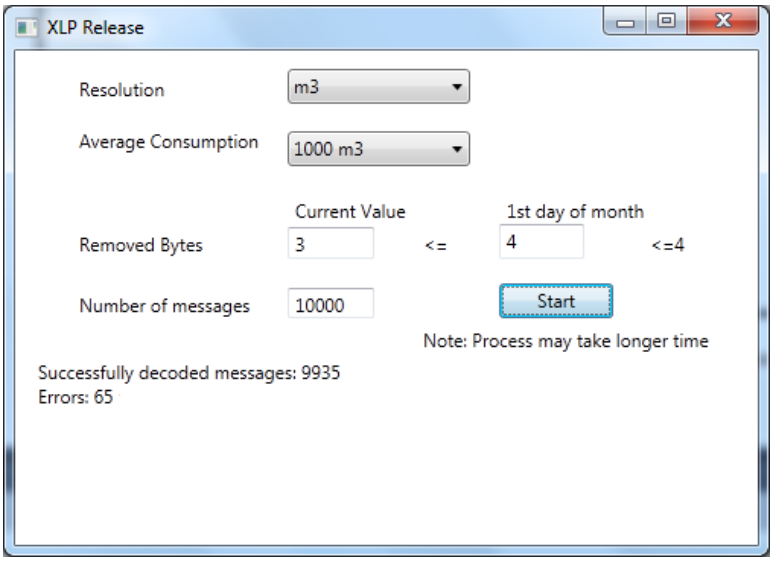

Figure 6.5: Screenshot of the simulation tool's GUI

6.2 Results

In this section are presented the observed results for each expected maximum consumption and resolution. In the cases when the amount of wrongly decoded messages became high enough, further data reduction was not performed since it is expected that the results will get even worse.

6.2.1 Expected water consumption of maximum $1000m^3$ and transmitted information down to 1 liter

Table 6.1 shows the results that were observed for such configuration. Since the initial tests showed good performance when 2 bytes of each data were removed, it was made a big test to make sure that it works properly. The results were satisfying, but for this simulation were needed ≈ 3 hours (laptop with $4GB$ RAM and 2.4 GHz Dual core processor)

Reduced bytes		Compared	Average	
$Data1$	$Data2$	Messages	Errors	Errors[%]
1	1	10 000	0	0
1	2	10 000	0	0
2	2	1 000 000	0	0
2	3	3000	909	30

Table 6.1: Results for max water consumption $1000m^3$ and resolution 1 liter

Reduced bytes		Compared	Average	
$Data1$	$Data2$	Messages	Errors	Errors[%]
2	2	10 000	0	0
2	3	10 000	0	0
3	3	100 000	0	0
3	4	30 000	230	0.8
4	4	100	93	93

Table 6.2: Results for max water consumption $1000m^3$ and resolution $1m^3$

6.2.2 Expected water consumption of maximum $1000m^3$ and transmitted information down to $1m^3$

In table 6.2 are presented the results for max consumption of $1000m^3$ and resolution down to $1m^3$. Again the most tests were performed in the case with maximum reduction and no errors. On the other hand, a relatively good result was observed when $Data2$ was not transmitted at all, but was "guessed" by the algorithm (less than 1%). The final test was made just from curiousity.

6.2.3 Expected water consumption of maximum $100m^3$ and transmitted information down to 1 liter

In table 6.3 are presented the results when max consumption is $100m^3$ and the resolution is down to 1 liter. This was the test with the highest jump in the errors after reduction of only 1 byte. This shows that the LSBytes are crucial when the resolution is big.

6.2.4 Expected water consumption of maximum $100m^3$ and transmitted information down to $1m^3$

Since the tests in the expected consumption of $1000m^3$ for resolution of $1m^3$ showed that it is enough to transmit only one byte of each value, so it was expected that for $100m^3$ will give the same result. For this reason it was tested the option with total brute force and trying to recover the

Reduced bytes		Compared	Average	
$Data1$	$Data2$	Messages	Errors	Errors[%]
2	2	100 000	0	0
2	3	100 000	0	0
3	3	250	151	60

Table 6.3: Results for max water consumption $100m^3$ and resolution $1m$ liter

Reduced bytes		Compared	Average	
$Data1$	$Data2$	Messages	Errors	Errors[%]
3	3	100 000	0	0
3	4	100 000	0	0
4	4	10 000	292	2.9

Table 6.4: Results for max water consumption $100m^3$ and resolution $1m^3$

message if only the CRC-payload is transmitted. The results (table 6.4) were surprisingly good.

6.3 Current consumption and Throughput

The reduction of the frame size is expected to improve the throughput probability and to reduce the current consumption. Let's assume that a message with the same data from the same meter is transmitted as XLP, Compact M-Bus and Standard M-Bus messages. The content of the data is the same as observed so far - current value, historical value, info codes. It is assumed that there is only current active codes, and no active ones in the past 30 days. On fig. 6.6 are presented the different frames and their lengths for each type (the preambles of the messages are also presented, since XLP reduces information also there). The investigated Frame Format is B, so there is only one CRC in the end of the frame (which is 2 bytes and is not shown on the figure). It should be noted that the encryption requirement for number of bytes multiple of 8 or 16 is not taken into account.

For the calculations is assumed:

- bit rate $Rb = 100kbps$, therefore transmission time $t = \frac{8 \cdot size}{Rb}$ ($size$ - number of bytes in one frame)
- transmission interval $T = 15s$
- number of meters $N = 1000$

Standard M-Bus: Total bytes - 41 (with CRC)

Preamb.	L-field	C-field	ID	CI	Header	Init.Vect	DRH1	Data1	DRH2	Data2	DRH3	Data3
8 byte	1 byte	1 byte	8 byte	1 byte	4 byte	2 byte	2 byte	4 byte	2 byte	4 byte	1 byte	1 byte

Compact M-Bus: Total bytes - 40 (with CRC)

Preamb.	L-field	C-field	ID	CI	Header	Init.Vect	FS	CRC-pld	Data1	Data2	Data3
8 byte	1 byte	1 byte	8 byte	1 byte	4 byte	2 byte	2 byte	2 byte	4 byte	4 byte	1 byte

XLP: Total bytes - 23 (with CRC)

Preamb.	L-field	ID	Header	Init.Vect	CRC-pld	Data1	Data2	Data3
7 byte	1 byte	2 byte	4 byte	2 byte	2 byte	2 byte	2 byte	1 byte

Figure 6.6: The same data presented in the three different frames - Standard M-Bus, Compact M-Bus and XLP

Frame Format	Size [bytes]	Transmission time [mS]	Probability 1000 meters [%]	Probability 10 000 meters [%]
Standard M-Bus	41	3.3	64.6	1.3
Compact M-Bus	40	3.2	65.3	1.4
XLP	23	1.8	78.2	8.6

Table 6.5: Probability for throughput for Standard M-Bus, Compact M-Bus and XLP messages

6.3.1 Packet Throughput

Every meter transmits periodically every T seconds. The transmission time is t seconds. To calculate a probability that a certain packet will not collide with other packets, we need to calculate probability that no other packets are sent by other transmitters t seconds before and t seconds after "our" packet transmission. Therefore, the vulnerable period is $2t$ seconds. Assuming that there are N transmitters and they are independent, this probability will be a product of individual probabilities.

The probability transmitter does not transmit during $2t$ seconds in a period of T seconds is $\frac{T-2t}{T}$, therefore the probability that a packet will not collide is:

$$P_{success} = (1 - \frac{2t}{T})^{N-1}, \tag{6.1}$$

where

In table 6.5 is presented the probability for throughput for each of the messages for 1000 and 10 000 meters in the coverage area.

The results show that if a lot of meters are in the coverage area, the distribution of the IDs will be the least problem for the designers of the system. This proves that transmitting only 2 bytes from the ID is enough for the purposes of "drive-by" data collecting. In these cases a drive-by method will be inefficient, so it should be searched approach with fixed network and transmissions in much bigger intervals.

Frame Format	Duty cycle %	Average consumption [μA]
Standard M-Bus	0.022	7.4
Compact M-Bus	0.021	7.3
7 XLP, 1 M-Bus	0.013	4.6
XLP	0.012	4.2

Table 6.6: Duty cycle and average consumption for the different frame formats

6.3.2 Current consumption

On fig. 6.7 is presented an estimation of the lifetime of a water meter when is supplied with battery which capacity $2000mAh$. The average consumption of the device without transmitting is considered $6.5\mu A$, in order to fit the values such that a meter transmitting compact M-Bus message to reach the limit after 16 years (which is reasonable, considering that Miitors' meters are estimated to consume less than $5\mu A$). In these $6.5\mu A$ are included the self consumption of the meter and self consumption of the RF chip in stand-by ($< 1\mu A$). In the simulation is taken also under consideration the fact that the transceiver switches from Stand-by to an IDLE mode before switch to Tx/Rx modes. The consumption in Idle mode is $1.3mA$ and the time for transaction $IDLE-Tx$ is $400\mu S$ [14]. On table 6.6 is presented the average consumption of a transmitter using the three different frame formats. It is considered that it transmits once every 15 seconds

6.4 XLP proposal for Miitors water meter

The results from the simulations showed that brute force based on CRC is relatively powerful tool for proper "guessing" of the unknown information up to a certain level. The current algorithm proposal is based on the assumption that the monthly water consumption is up to $1000m^3$ and the meter transmits the values with resolution 1 liter. Thus, the reduction is 2 of the MSBytes from each of the two values for the water consumption in order to assure 100% accuracy. On fig. 6.8 and 6.9 are presented block diagrams of the algorithms for encoding and decoding such XLP message. The indexes of $DataField[]$ and $XLP[]$ are exact numbers, calculated for this algorithm. It is made this way for two reasons: 1) To give the reader more acurate picture of how the message is processed byte by byte, and 2) to show the importance of the requirement that the receiver must know how many bytes are removed and from where. In case of other requirements for the water meter (regarding monthly consumption and resolution), these indexes should be changed in order to fullfil the correct structure of the messages.

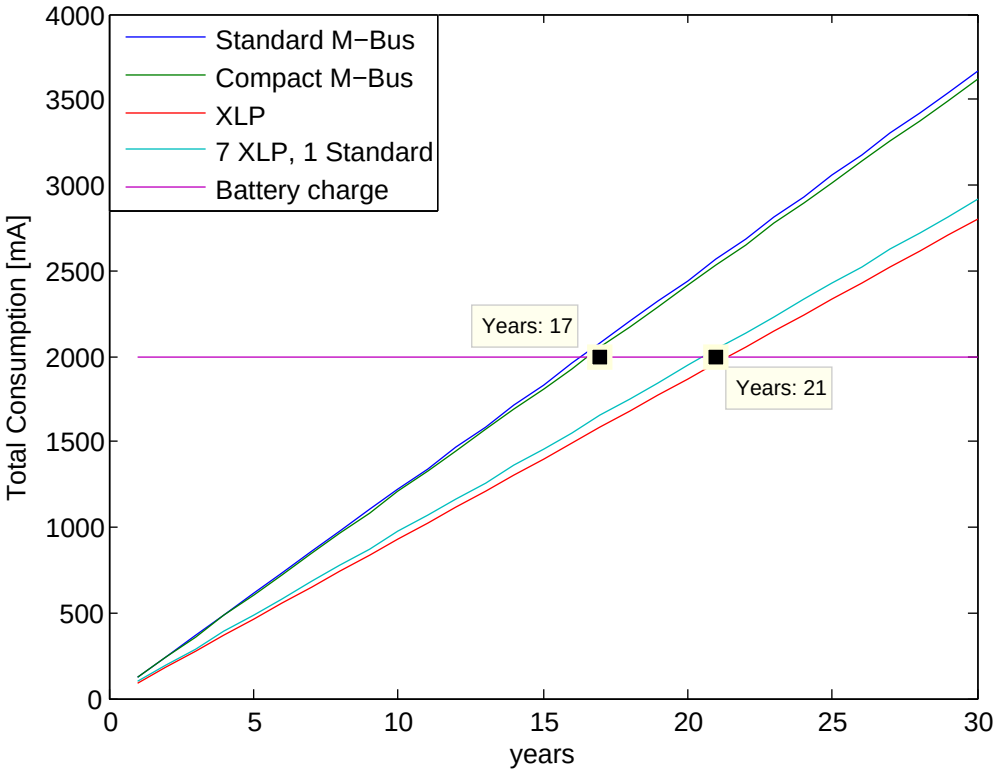

Figure 6.7: Average current consumption for the different frame formats

6.4.1 XLP encoding

The algorithm starts right after standard conditions are detected ($C = 44_h$ and $CI = 7A_h$). First is checked if a default message will be transmitted (Current value, Value for the 1^{st} day of the month and info codes). If so:

- a flag is raised ($STS[5] = 1$: bit 5 of the Status byte, which is part of the header);
- the Data Field is a concatenation of the data for CRC-payload and 2 of the LSBytes from each value;
- in case that there is active info code in the moment of the transmission, the byte with current info codes is added to the Data Field.
- in case there have been active info codes in the past 30 days, in order the receiver to be able to interpret the message correct, to the Data Field are added the byte with these info codes AND the byte with the current active ones (even if currently there is no active one). If there have been active more than 1 info codes, this data should be filled in order of their active codes, presented in the info code byte (e.g. $info30 = 10010000$, then the following information is: first

$info30[0]$; then $info30[3]$). The filling of the info codes should start from the MSB, since otherwise, there is option that an info code byte get value 00101111, which is $2F_h$ and the receiver will interpret that as a "dummy" byte. For this reason, LSB should be 0 always (or some other one from the first 4 LSB)

If the message is not with a default content, the Data Field should start with a Format signature (2 bytes calculated on the DRH fields, as in Compact M-Bus Frame, or 1 byte based on an already filled table with possible modes of the meter, as suggested in section 5.2.7, depending on which strategy for the meter is chosen). Then the length of the frame L is calculated (Note that ID is the 2-byte part of the full ID of the meter; the *header* is 4 byte and the $InitVect$ is the Initialization Vector of the encryption, which is $2F_h2F_h$ for AES-128, or 2 timing bytes for DES). Then XLP message is created. Before the transmission, a CRC is calculated over the whole frame and is added to the XLP message.

6.4.2 XLP decoding

After a XLP message from a meter subscribed in the receiver's database is received and decrypted, starts the algorithm for decoding the XLP data and recovering the full M-Bus message. The M-Bus frame fields are filled with the known data: C-field is 44_h, CI-field is $7A_h$, the header is 4-byte short header and the following 2 bytes from the initialization vector. After, starts the decoding of the Data Field. If $STS[5] = 1$:

- the corresponding bytes of the Data Field are filled with the DRHs for the current and historical ($1^{st}day$) values;
- for each value is allocated 4 bytes of data. The first 2 bytes are filled from the received message and the rest is taken from the last record, saved in the database;
- if the XLP message is not finished yet ($L < 14$) and the following bytes are not "dummy", then in the data field is added new DRH (DIF3), which shows that until the end of the message there will be manufacturer specific data structure. In this case, the info codes;
- if there is only one byte left then this means "There are current active info codes, but none of them has been active in the past 30 days";
- if the bytes are more, the first one is "current active info codes", then "active codes in the past 30 days", followed by information about the time they have been active. The interpretation of this data should follow the same pattern as the way it was produced (explained in the algorithm for XLP encoding)

- After the message is recovered as a standard M-Bus , a CRC is calculated over the Data Field and is compared with the CRC-payload that has been transmitted with the XLP message ($XLP[9, 10]$). If they are equal, the message is considered as decoded. Otherwise, the data taken from the database is incremented according to the algorithm presented in section 6.1 and is compared again until they become equal

In case $STS[5] = 0$, the first 2 bytes of the Data Field are Frame Signature. Then the receiver has to check if this FS is present in its database. If yes, the data is interpreted in according to the algorithm corresponding to this FS (in case the data is with reduced bytes, the same recovering procedure can be implemented). If not, there are two options, depending on the strategy the company has chosen for this water meter:

- If the meter is programmed to transmit a full M-Bus message after certain interval, the receiver has to wait until a full M-Bus message from this meter is received.

- If not, the message is discarded.

6.4.3 Encryption

The requirement for the encryption to have encrypted data with number of bytes multiple of 8 or 16 brings some complications in the efficiency of the protocol, since the data should be filled with "dummy" bytes (see section 2.3.3). The algorithm for default message, presented in this section, has $InitVect + CRCpay + Data1 + Data2 = 2 + 2 + 2 + 2 = 8$ (if no info codes are or have been present). In case that the manufacturer choose minimum transmission time, then a partial DES encryption should be implemented (for the first 8 bytes). Then the values of the water meter will be encrypted and the data for the info codes will not.

Though, DES encryption is considered as not very reliable and for wM-Bus messages it is recommended AES-128. In this case the encrypted data should be with number of bytes multiple of 16, which stultifies the reduction of the data values.

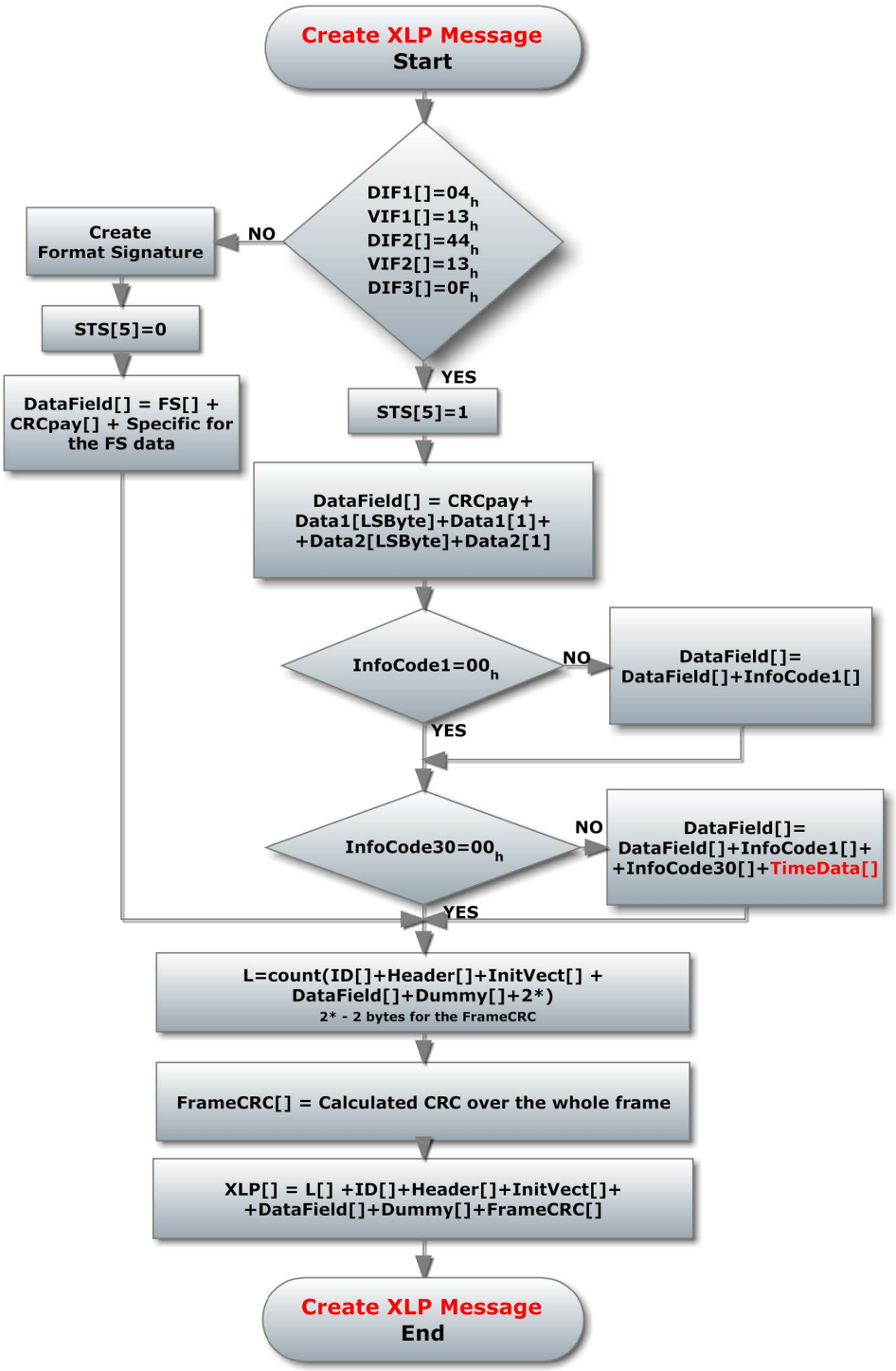

Figure 6.8: Algorithm for encoding an XLP message

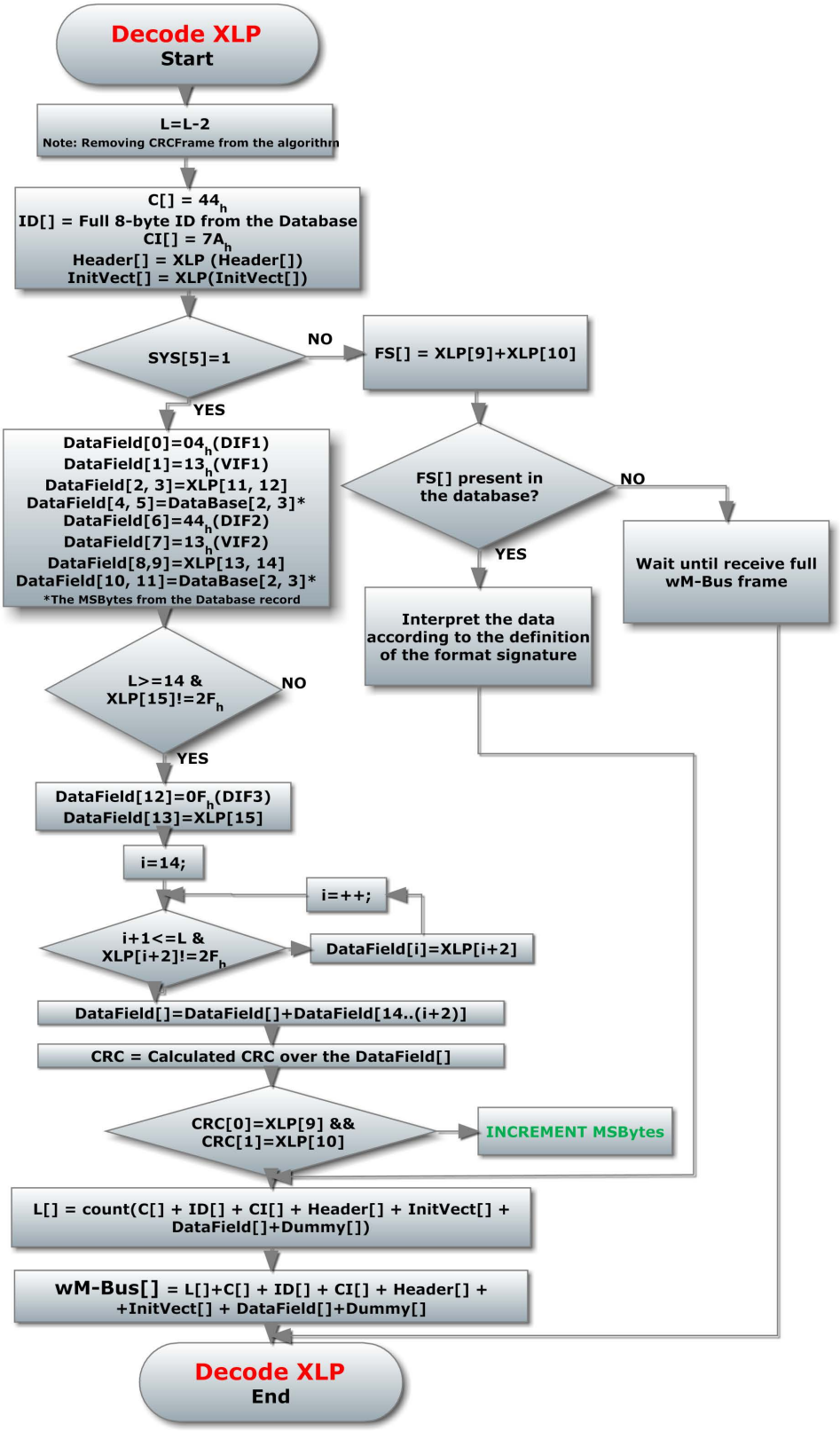

Figure 6.9: Algorithm for decoding XLP message

Chapter 7

Conclusions

In this project was investigated the possibility of creating a frame format for eXtremely Low Power wireless communication based on Wireless M-Bus. The results showed that despite the fact that this protocol is created with the main purpose to be very power efficient with respect to radio transmissions and computing power, some aspects can be improved for the cost of others. The implementation of XLP frame format in the battery powered devices which transmit frequently can prolong the battery life with several years. Furthermore, the short transmission time can open doors for supplying the devices with alternative energy harvesting sources.

It was explored the possibility to use Cyclic Redundancy Check not only for detecting errors occurred during the transmission, but also as a tool for recovering information. The simulations showed that such use of CRC is possible, at least for short messages. Though, this requires certain computational power in the data recovering device (in this case in the receiver).

Additionally, a result of this project was development of simulation tool for investigating the results of data reduction, which is recovered with the help of CRC. Currently it works for very narrowed use cases, but with slight modification based on the requirements of the messages, it may be used for simulation of additional data reduction/recovery.

Chapter 8

Future Work

There are several aspects, proposed in this report, that have not been investigated:

- To be implemented "merging" of the timing information for the active info codes (mentioned in section 4.3.1);
- Dynamic data reduction based on the consumption for the last 30 days;
- Optimization of the decoding algorithm for higher accuracy and speed;
- Optimization of the decoding algorithm for different time lines (data in the database to be newer than 1^{st} day of the month data). This could be implemented by including observation of the date in the database and the algorithm.

With respect to a future work that can be observed as project by itself, there are several different aspects.

8.1 Improving the Encryption Model

Currently it is suggested an encryption AES-128 which uses initialization vector based on meter's address and identification bytes and access number byte. This provides certain vulnerability considering that there is limited number of unique vectors. An option is transmitting information with the operating time of the meter and use it as an initialization vector. This will improve the robustness of the encryption.

8.2 Identifying Transmitting Meters

Currently it is assumed that all the meters in the coverage area of the receiver have unique addresses. It is possible to reduce the transmitted ID bytes even more, which will lead to existence of several meters with the same ID

configuration. Based on the fact that the meters have static positions, it is possible to implement a solution for accurate identification of specific meter based on:

- GPS coordinates of the receiver
- GPS coordinates of all the meters
- RSSI from every meter
- considering that every meter transmits in strict intervals, the receiver can add time stamps to all the received messages.
- strictly defined route of the receiving vehicle

It is also possible to include in the identifying the transmitting meter a hint based on the data it sends (if 2 meters with the same ID provide information 45 liter and 23 liters, and if according to the last data collecting one of them has had 40 liters, it is not likely that the value 23 is from it). This faces a problem that if the information is encrypted, there is no access to the real values of the data. There could be investigated certain criteria that is is possible to decode and define the transmitter more accurate.

8.3 Batteryless operation

The suggested protocol is very low power consuming, but despite that, there is still requirement for presence of battery in the metering device. With the development of the technologies, number of energy "harvesters" were created in the past years, which collect energy from the environment they are located in.

An interesting work could be investigating of using an alternative power source. A major requirement for such source is to be able to provide current $\approx 35 - 40mA$ in relatively short time $(2 - 3mS)$.

8.4 Fixed Network

With the developing of the routing protocols for Wireless M-Bus, a particularly interesting future work will be investigating of implementation of a fixed network based on this technology. On fig. 8.1 is presented simplified possible scenario. The sensor nodes (meters) use XLP protocol for communication with their corresponding cluster nodes (data collectors or routers), which form a mesh network in order to transmit the information to the water provider base, where the data will be processed for future exploitation.

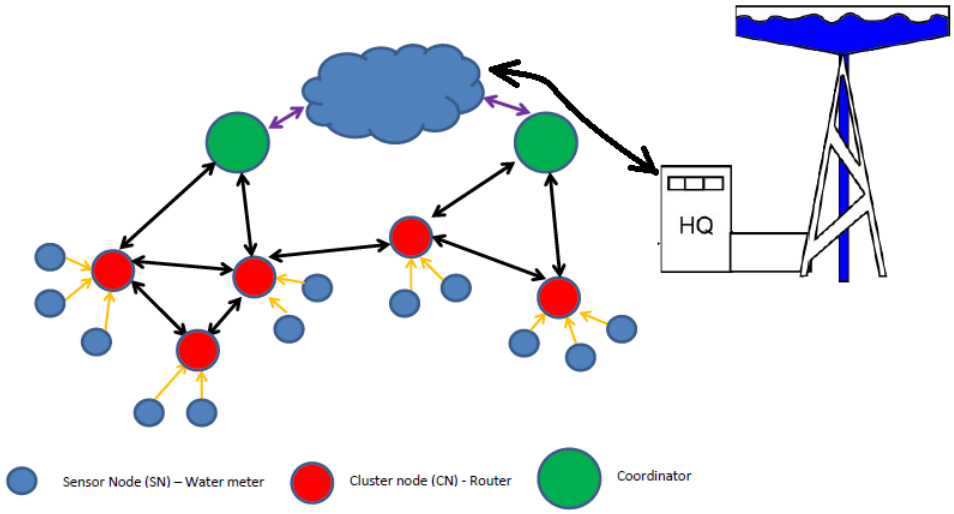

Figure 8.1: Future Work Scenario

Bibliography

[1] M-bus. http://www.m-bus.com/.

[2] Tomas arnewid, 2009. Presentation: Gøteborg – The first ZigBee City?

[3] Statistics Denmark. Bil3: Number of households by region, use of cars and household type. http://www.statistikbanken.dk.

[4] EN 13757-1. *Communication systems for meters and remote reading of meters - Part 1: Data exchange*, 2002.

[5] EN 13757-2. *Communication systems for meters and remote reading of meters - Part 2: Physical and link layer.*

[6] National Instruments. Zigbee. http://zone.ni.com/devzone/cda/tut/p/id/7118#toc0.

[7] Kamstrup. *Miltical 21 - Datasheet.*

[8] John McNabb. Vulnerabilities of wireless water meter networks. *Black Hat USA Las Vegas*, 2011.

[9] Netbeheer Nederland. *P2 Companion Standard - Dutch Smart Meter Requirements*, March 2012.

[10] prEN 13757-3. *Communication systems for and remote reading of meters — Part 3: Dedicated application layer*, 2010.

[11] prEN 13757-4. *Communication systems for meters and remote reading of meters - Part 4: Wireless meter readout (Radio meter reading for operation in SRD bands)*, 2011.

[12] prEN 13757-5. *Communication systems for meters and remote reading of meters - Part 5: Wireless Relaying*, 2009.

[13] Citygrow Energy Systems. Advanced metering infrastructure. http://www.citygrowsys.com/zaspx/products.aspx?id=105.

[14] Texas Instruments. *CC1121 - High Performance Low Power RF Transceiver (Datasheet).*

Appendix A

3-out-of-6 encoding

On the figure below is presented the table of the 3-out-of-6 encoding for T-mode wM-Bus protocol. As can be seen, every 4-bit symbol is presented with 6 bits with equal number of 1s and 0s.

NRZ-Code	Decimal	6-bit code	Decimal	N° of transitions
0000	0	010110	22	4
0001	1	001101	13	3
0010	2	001110	14	2
0011	3	001011	11	3
0100	4	011100	28	2
0101	5	011001	25	3
0110	6	011010	26	4
0111	7	010011	19	3
1000	8	101100	44	3
1001	9	100101	37	4
1010	10	100110	38	3
1011	11	100011	35	2
1100	12	110100	52	3
1101	13	110001	49	2
1110	14	110010	50	3
1111	15	101001	41	4

Figure A.1: 3-out-of-6 encoding for T-mode wM-Bus protocol [11]

Appendix B

T2-mode

On figure B.1 is presented an example T-mode of 2-way communication. The "short reception window" is usually $3mS$.

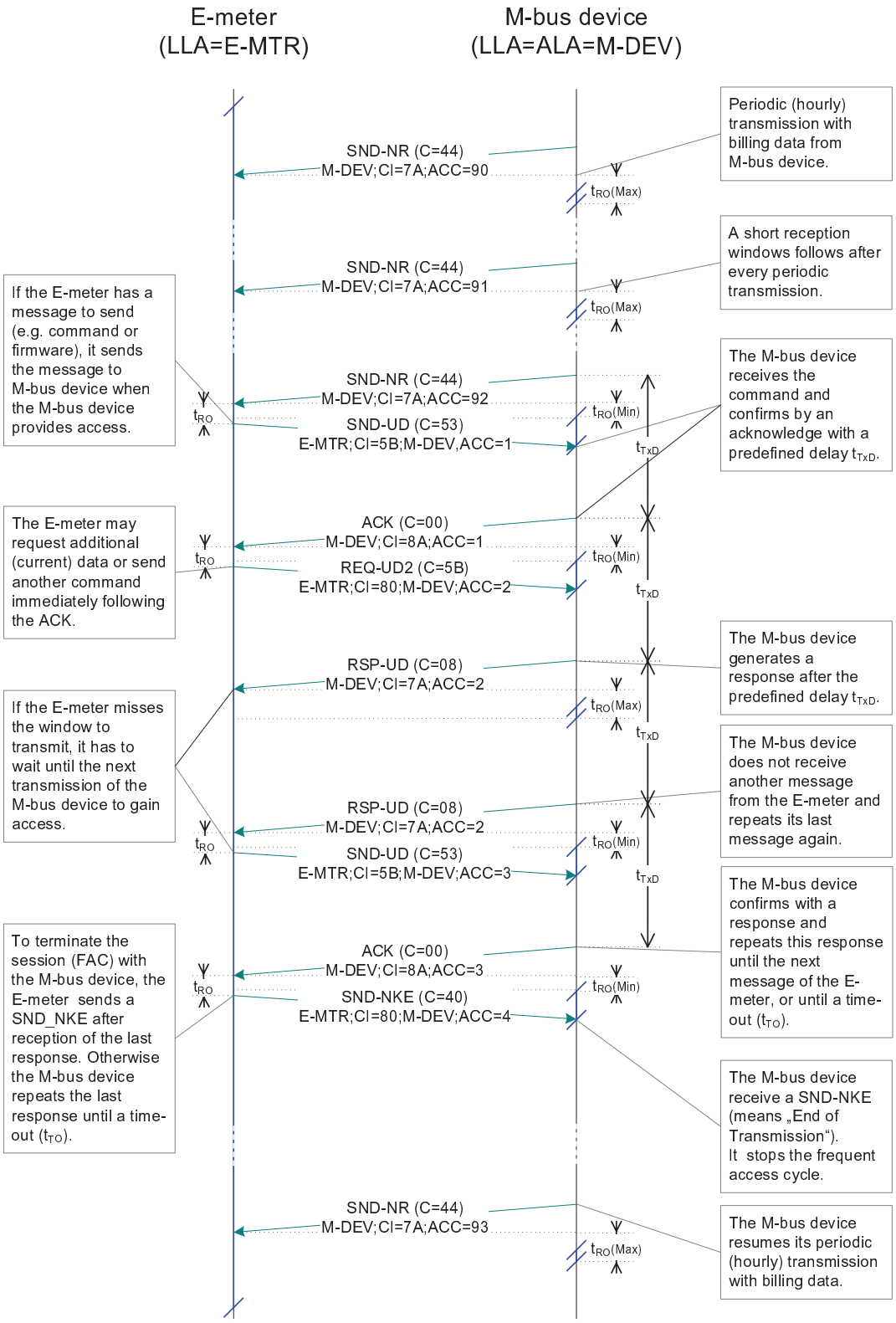

Figure B.1: Mode T2 [9]

Appendix C

CD with the Simulation tool and Matlab codes

To this report is attached a CD with the simulation tool used in this project. It is provided an executable file with and the source codes. It is also attached Matlab codes used for some of the calculations in the reports.