Autonomous Temperature Sensor for Smart Agriculture

Wireless communication & Networking

A.J.R. Bleeker

R. Wijnands

TU Delft Delft University of Technology

June 2020

Autonomous Temperature Sensor for Smart Agriculture

Wireless communication

by

A.J.R. Bleeker, 4713230

R. Wijnands, 4646738

Delft University of Technology

Faculty of Electrical Engineering,
Mathematics and Computer Science

In partial fulfilment of the requirements for the degree of

Bachelor of Science
In Electrical Engineering

To be defended in public on 29 th June 2020

Supervisors:	Dr. Q. Fan	TU Delft
	Ing. R.M.A. van Puffelen	TU Delft
	MSc. L. Pakula	TU Delft

Abstract

In this thesis, the design and implementation of a wireless communication module and an accompanying network are discussed. This wireless communication module is used in a device which communicates frost temperatures measured at orchards. The goal is to gather data on how fruit frost develops on orchards, and to warn a fruit farmer if it starts freezing. First, several wireless communication protocols are discussed and the one best suited for the application is chosen. It is decided to use the LoRaWAN communication protocol. Then, the off-the-shelf hardware components which are required to implement the LoRaWAN communication protocol are chosen. Furthermore, LoRa communication parameters and the LoRaWAN network structure are discussed. A scheduling system is designed and proposed to increase the reliability of the network. At last, network simulations are performed to verify the chosen implementation.

Preface

First of all, we are very grateful to Ing. R.M.A. van Puffelen, MSc. L. Pakula, and Dr. Q. Fan for guiding this project. During the past few weeks, they were always available for help and guidance. Furthermore, we would like to thank everybody involved in the organization of the Bachelor Electrical Engineering and the Bachelor Graduation Project.

As a group of six students, we chose this project due to the wide variety of Electrical Engineering aspects included. Every single student was able to find a subject within the design of the autonomous sensor for smart agriculture, in which he could express his academic skills. Also, a nice property of this project is that besides a theoretical side, also many practical aspects of designing such an autonomous smart system are covered. Although this project is completed during the COVID-19 pandemic, and thus mainly consist of theoretical system design and simulations, it still was a very educational and interesting project to complete from our home base. We hope you can appreciate and enjoy reading this thesis.

Arnoud Bleeker

Ruben Wijnands

June 2020

List of abbreviations

ABP	Activation By Personalization
ACK	Acknowledgement
API	Application Programming Interface
BW	Bandwidth
CR	Coding Rate
CSS	Chirp Spread Spectrum
DR	Data Rate
EIRP	Effective Isotropic Radiated Power
ERP	Effective Radiated Power
LPWAN	Low Power Wide Area Network
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Media Access Control
MCU	Microcontroller Unit
OTAA	Over-The-Air Activation
PCB	Printed Circuit Board
PHY	Physical Layer
QoS	Quality of Service
RF	Radio Frequency
SF	Spreading Factor
TOA	Time On Air
VSWR	Voltage Standing Wave Ratio
WSN	Wireless Sensor Network

Contents

Chapter 1

Introduction

In the Netherlands fruit production is a large segment of the economy. There are over 2600 fruit production companies which together cover over 20,000 hectares of ground [1]. Most of these Dutch companies focus on the production of pears and apples. The fruit production in the Netherlands, but effectively all around the world, faces a large problem related to spring frosts in fruit trees. If temperatures drop below the critical temperature, open and blooming flowers of the fruit trees can be damaged causing less fruit to grow. The frost damages are mainly caused by the formation of ice. Intra-cellular ice formation breaks the blossom's tissue structure and causes a cell death [2]. Freeze injury is nowadays the biggest problem of fruit production, which causes a loss far greater than any other type of natural hazard encountered with the production. As a result, the yield of production and distribution of fruits are restricted.

A lot of research has been done to obtain a reduction in the losses caused by spring frosts. Two of the main solutions that are provided are frost protection with sprinkler irrigation and frost protection with wind machines. The frost protection with sprinkler irrigation works using extra-cellular ice formation to prevent intra-cellular ice formation. Sprinkling water onto the tree's flowers and buds causes ice nucleation on the outer surface. This then causes the freezing of the water transporting vessels which protects the flowers due to gradual dehydration [3]. Frost protection with wind machines aims to prevention of intra-cellular ice formation. By using a large wind machine or rotating fan a light wind (1.5 m/s) is created which causes an instantaneous increase in temperature [4]. Because of the fan, a temperature increase is obtained of up to $1\ °C$ at a 15m distance from the wind machine. Due to this small increase, the flowers are protected from the frost.

So now there are two effective solutions used to prevent spring frosts in fruit trees. Both of these solutions have a requirement to know the temperature of the air surrounding the trees, and their control systems make decisions based on these measurements. But to implement these solutions in large scale fruit production companies, an accurate temperature measurement is required over the whole field, rather than one single temperature measurement. Therefore this project focuses on the acquisition of the 3D temperature profile of a fruit orchard in the temperature range near the critical temperature, which is the temperature where the fruit buds are damaged.

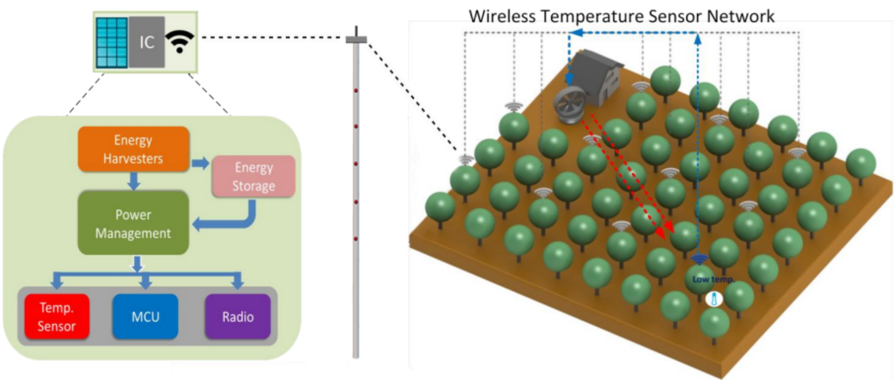


Figure 1.1: A general system overview.

The great advantage of the acquisition of the temperature profile in contrast to single temperature measurement is that it provides the ability to perform local frost protection rather than frost protection over the whole field. The structure of such a local frost protection system is illustrated in Figure 1.1. Local frost protection in turn provides a reduction in the use of resources such as water and electric energy. In addition, the acquisition of the temperature profile provides a valuable resource for further research on the effectiveness of frost protection methods.

To realize this large scale sensor grid, end-devices are created which have two main design requirements. They have to transmit their gathered data wirelessly and harvest their own energy. To meet these requirements, the system design is split up into three design challenges. Namely, energy harvesting and control, measurement and control, and wireless communication and networking. This thesis focuses on the wireless communication and networking part of the autonomous smart temperature measuring system. However, before the focus is put on designing this subsystem, first an overall system design is created such that all subgroups have a clear end goal in mind.

The sensor network consists of multiple sensor end-devices. This end-device includes an energy harvesting module, a controller, a wireless communication module, and five temperature sensors, all integrated onto a pole. These poles are positioned among the trees of the fruit orchard, and together they form a smart wireless temperature sensor network in the fruit orchard. To get a rough idea of this design an example is given in Figure 1.2. In Appendix A, the choice for this implementation containing one pole with one energy harvesting module, one controller, one wireless communication module, and 5 temperature sensors is explained.

Figure 1.2: Overview the designed system in the orchard

Chapter 2

Program of requirements

In the Netherlands large amounts of fruit harvest are lost due to freezing of the flower buds. To find a solution for this problem, atmosphere scientists are in need of a system to monitor the air temperature on fruit orchards. To make good measurements it is required that not only the temperature is measured as a 2D position on the orchard, but also the temperature is measured in terms of height, such that a 3D temperature map of the orchard can be made. If a good solution is found, the build monitoring system should be used for further monitoring of fruit orchards as a commercial product.

2.1 Requirements for the entire system

The requirements for the entire system are subdivided into functional requirements and non-functional requirements. Functional requirements are subdivided into mandatory requirements and trade-off requirements. Mandatory requirements are defined as the requirements the product must always comply with to consider the design acceptable. Trade-off requirements are defined as the requirements which are preferred and make end-users increasingly satisfied.

2.1.1 Functional requirements

Mandatory requirements
The system must

- have an energy harvesting unit, that is capable of sustaining all power requirements of the system, from ambient energy sources
- contain an energy storage unit, that is capable of sustaining the continuity of the power delivery
- have wireless communication network to share the measured data with the end-user
- have end-devices with a local smart processing element
- provide a 3D temperature profile of the fruit orchard, with 5 sensors each placed at a different height
- have data storage of measurements
- be capable of performing adaptive temperature measurements
- have a temperature measurement accuracy of at least 0.5 °C of the absolute temperature
- be able to warn the farmer when frost prevention action is required

To create a 3D temperature map of the orchard, temperature measurement has to be done at several points in the field. Each measurement has to be stored at a central data location. This means a data communication network is needed, which enables each temperature sensing end-device to send its data to this central point. To reduce the hassle and cost of laying down a cable to each end-device, the assignment requires that the end-devices should be able to transmit their gathered data wirelessly. The fact that no external cabling can be used also means all the power needed, to gather and send the temperature data, should be contained in each end-device. Another requirement of the assignment is that no usage of long term battery storage is allowed, where the battery should be replaced after some time. Therefore, the system should harvest and store the energy it needs to perform critical tasks. To ensure the temperatures which are of most interest are measured, the amount of performed measurements should be adaptive. The system should not measure and send high-temperature data as frequently as low-temperature data near the critical temperature at which crops might get damaged.

Trade-off requirements
The system might

- have bidirectional communication
- have system failure self-diagnosis
- be able to update firmware remotely
- have energy level monitoring
- have a power saving mode

2.1.2 Non-functional requirements

- The total cost of an end-device should be minimized in the order of €100
- The system must support one end device with 5 sensors at different heights per 100 m^2
- The system should be able to cover at least 10 hectares
- The system should have a lifespan of at least 20 years in normal operation conditions

The average size of a fruit orchard is 8 hectares in the Netherlands. However, in France the average size of a vineyard is a little higher than 10 hectares [5]. It should be noted that the above-stated system requirement regarding the area to cover is a minimal requirement since the area of the orchard is only the average.

2.2 Requirements for the wireless communication system

- The communication range should be at least in the order of 1 km
- The network must be able to support up to 2000 nodes
- The network should be able to adapt the critical temperature value in an end device, such that different sort of crops can be supported
- The network must support the ability for each end-device to send a message at a maximal frequency of once every 5 minutes
- The network should be able to change the end-device operation mode for off-season periods
- The network should have proper reliability concerning data message delivery
- The network should abide to European regulations and laws

Since the average size of a fruit orchard is 10 hectares, the minimum range is approximated by a worst-case situation. First, it is supposed that the orchard can scale up to 20 hectares. In a worst-case situation, the orchard has a length/width ratio of 1:5. Using an orchard of 20 hectares with a ratio of 1:5, the maximum range that should be obtained is approximately 1 km. Having a worst-case scenario of 20 hectares and a maximum node density of 1 per 100 m^2 means the maximum amount of nodes is 2000. As explained by the measurement and control group [6], a transmission rate of once every 5 minutes is required to effectively detect a maximum temperature change of approximately 1 °C. Furthermore, the warning capability of the designed system is not required when the fruit blossom is not present. The period in which fruit blossoms are not present and thus the crops do not need protection is referred to as the off-season period. This off-season period should be configurable by the fruit farmer.

Chapter 3

Wireless communication protocol

Following the requirements, a wireless communication network is necessary to share measured temperature data with the orchard farmer. Also, the orchard farmer must be warned if frost occurs. Besides that, it is desired that all measured temperature data is stored at a central location. The first step in designing and realizing a wireless communication network is choosing the right communication protocol for the end-devices. Several wireless communication protocols are compared and the best fitting wireless communication protocol, fulfilling the requirements for this application, is chosen.

3.1 Feasibility of investigated technologies

In this section, the feasibility of several wireless communication protocols is investigated. Based on the requirement of energy harvesting from ambient energy sources, low power consumption of end-device is required. A narrow down is made regarding the number of possible wireless communication protocols. Then, following the requirement of deploying a large scale Wireless Sensor Network (WSN), protocols with a lack of large scale support are directly omitted. Finally, based on the availability and support of certain wireless communication protocols, the remaining possible WSN technologies are presented and compared in Section 3.2.

3.1.1 Low power communication

Since the end-devices deployed in the orchard are powered from ambient energy sources only, power consumption is required to be minimized. Wireless communication protocols such as WiFi or 4G are not considered suited since their power usage is too high for this application. Therefore, only Low Power Wide Area Network (LPWAN) and Low-Rate Wireless Personal Area Network (LR-WPAN) technologies are investigated in Section 3.1.2.

3.1.2 LPWAN or LR-WPAN

In the present day, almost all low power wireless communication protocols fall in either one of two categories, namely LPWAN or LR-WPAN. Both of these categories achieve low power usage by having a low data rate. LPWAN is specifically designed for long range communication, in the range of 5 km to 50 km [7]. Typical examples of LPWAN technologies are LoRaWAN and SigFox. Examples of LR-WPAN technologies are ZigBee, 6LoWPAN, and Bluetooth Low Energy which are made to operate in a range of 10 m to 100 m. This means that using LR-WPAN technology requires the use of a mesh network to still cover the desired range of the fruit field.

Mesh network structures are not considered as a suitable solution for this problem. A mesh network requires all end-devices to be turned on to forward messages through the network. However, keeping the devices constantly listening for messages causes a large energy overhead. Also, end-devices close to the base station will receive and forward more packets than other end-devices which may cause a large difference in power usage. Failure of these end-devices might cause the entire network to get disconnected. Thus, it can be concluded that an LPWAN is most suited for this application.

3.1.3 Availability and support

Some widely used LWPAN technologies are LoRaWAN, SigFox, and NB-IoT. However, other protocols in the field of LWPAN exist such as DASH7 and Weightless. However, there is a lack of online documentation and support regarding these technologies, which makes comparing them difficult. It would be preferable to test these wireless communication protocols. Since during this project no physical testing is possible, these wireless communication protocols are not compared. However, it must be noted that this does not make these technologies less suited for the application.

3.2 LoRaWAN, SigFox and NB-IoT

Three LPWAN technologies LoRaWAN, SigFox, and NB-IoT, nowadays widely used, are investigated more elaborately on the following performance metrics: coverage and range, power consumption, scalability, deployment and cost, data throughput, security, and reliability.

3.2.1 Network coverage and range

The first metric to consider is the potential communication range a certain technology allows for. The exact range of the technology heavily depends on hardware implementation and environment. Thus, it is difficult to give an exact number, but it is generally regarded that NB-IoT has a maximum range between 1 km to 10 km, LoRaWAN between 10 km to 20 km, and SigFox between 10 km and 50 km [8] [9] [10]. Having a large range results in having better coverage per central receiving point, also referred to as a gateway. This means that for the SigFox network fewer gateways are needed to ensure coverage. NB-IoT has the advantage that it can leverage the existing 4G network of the national network operators in each country. In the case of The Netherlands, both SigFox and NB-IoT have complete coverage. However, looking at the entire European region, not every region is well covered [11] [12].

Public networks implementing LoRaWAN are available in different countries. However, these solutions have the same problem as with the networks of SigFox and NB-IoT, coverage is not guaranteed. What is different from the other two techniques is that using LoRaWAN enables the possibility to set up an entire private network and install an own gateway and network server without the need of a network provider. This guarantees coverage which is a huge advantage.

3.2.2 Power consumption

Power consumption of NB-IoT is relatively high due to frequent synchronization, Quality of Service (QoS) handling, and OFDM/FDMA access modes. This results in shorter battery life than devices based on LoRaWAN [13] [8]. SigFox packets have a relatively high time on air compared to LoRaWAN. This is due to the low data rate of SigFox, which is 100 bps. A SigFox uplink packet size ranges between 14 to 29 bytes. This approximately would take 1 to 2 seconds to transmit a message. LoRaWAN on the other hand has an adaptable data rate ranging from 0.3 kbps to 50 kbps. A LoRaWAN package length consists of an overhead ranging between 12 and 28 bytes and a payload ranging between 0 and 222 bytes, resulting in a lower time on air than SigFox in general and thus the power consumption is lower.

3.2.3 Scalability

SigFox, LoRaWAN, and NB-IoT all offer the support of a massive number of connected devices. However, NB-IoT allows for connectivity up to 100 K end-devices per cell against 50 K per cell of LoRaWAN and SigFox [8]. Furthermore, due to regulations LoRaWAN and SigFox have a limitation on the amount of data messages that can be sent on a day. NB-IoT has no limitations concerning the amount of data messages that can be sent and is thus more scalable. Because LoRaWAN and SigFox use the unlicensed 868 MHz, they are more sensitive to end-devices of other users of the spectrum. LoRaWAN performs significantly worse for a growing number of end-devices [14]. However, LoRaWAN easily supports thousands of end-devices when configured correctly in the operating range of interest ($\sim$ 1 km).

3.2.4 Deployment and cost

When considering the cost-effectiveness of the wireless technologies, NB-IoT and SigFox come with recurring subscription costs. As stated before, LoRaWAN has the ability to deploy a private network structure to circumvent network subscription cost, but this requires the deployment of an own gateway and network server.
Network subscription costs for SigFox are in the range of €6 to €20 per device per year, depending on the number of messages per day, ranging between 2 to a maximum of 140 messages. Network subscription costs for NB-IoT on the T-Mobile network are approximately €9 per device per year. For LoRaWAN, the subscription costs for The Things Network are approximately €4 per device per year. Also, there are differences in Radio Frequency (RF) modules. For NB-IoT, the Quectel BC66 RF chip has a cost of approximately €10 and the SARA-R4 series is approximately €20 per RF chip. For LoRaWAN chips, only made by

the company Semtech, the prices are around €3 to €5 for an RF chip of the SX126x and SX127x families. The prices for SigFox RF modules range from €1 to €2 per RF chip.

3.2.5 Data throughput

Concerning data throughput, it should be noted that no extremely high data throughput is required since the message payload is relatively small with only a few measured temperatures and possibly some other parameters such as the energy level of the end-device. SigFox has a limitation of 140 uplink and 4 downlink messages per day because of network operator regulations.
LoRaWAN messages must comply with a 1% duty cycle by ISM band regulations [15] and have a maximum payload of 222 bytes at a data rate of 0.3 kbps to 50 kbps. These data rates can change depending on the distance to the gateway, directly influencing the data throughput.
NB-IoT has no limitations regarding band usage and can offer a maximum payload of 1600 bytes [8]. NB-IoT claims high data rates of 200 kbps uplink and 180 kbps downlink, but this is not the effective data rate. Generally, peak uplink/downlink data rates of 66/27 kbps, respectively, are more representative, because acknowledgements in between the transmission packets are required.

3.2.6 Reliability

Reliability assessment is important since it is desirable to have a network that is running continuously and has a low packet loss. First of all, SigFox has limited reliability due to almost one-way communication since almost no acknowledgements are possible [16]. LoRaWAN also has limited bidirectional capabilities. LoRaWAN uses Aloha to access the medium and has no collision avoidance techniques, but makes use of orthogonal messaging to prevent packet loss [17]. However, NB-IoT is a synchronous protocol and does take care of message loss. Also, NB-IoT uses the HARQ mechanism that increases message delay to avoid data loss [9]. It can be said that the Chirp Spread Spectrum (CSS) modulation of LoRaWAN can handle interference, multipath, and fading, but it cannot offer the same QoS as NB-IoT [13].

3.2.7 Security

Another important performance metric is security since the system should not fail due to a lack of security. The security of NB-IoT is very high since it has LTE encryption [18]. LoRaWAN uses 128 bit AES security and SigFox does not support encryption [16].

3.3 Choice of wireless communication protocol

In Table 3.1, a decision table is made based on the evaluated performance metrics. The scores range from 1 which is the worst possible performance to 10 which is the best possible performance. From this table, it follows that LoRaWAN is the most suited wireless communication protocol for this project.

Table 3.1: Decision making matrix for wireless communication protocols

		LoRaWAN		SigFox		NB-IoT	
Metrics	Weight	Score	Weighted	Score	Weighted	Score	Weighted
Coverage and range	10	10	100	7	70	7	70
Power consumption	8	9	72	5	40	4	32
Scalability	6	7	42	8	48	10	60
Deployment and cost	6	10	60	6	36	3	18
Data throughput	4	6	24	3	12	8	32
Reliability	4	7	28	4	16	10	40
Security	2	7	14	1	2	10	20
			340		224		272

Chapter 4

Hardware components

It has been found that the LoRaWAN wireless communication protocol best suits this application, mainly due to the low power, low cost, and large coverage capabilities. Now, the necessary hardware to implement such a wireless network is investigated. First, the Wireless Sensor Network (WSN) structure is elaborated. Then, the individual hardware components which are vital to the WSN structure are recommended or chosen.

4.1 LoRaWAN structure

First of all, the LoRaWAN structure is visualised in Figure 4.1.

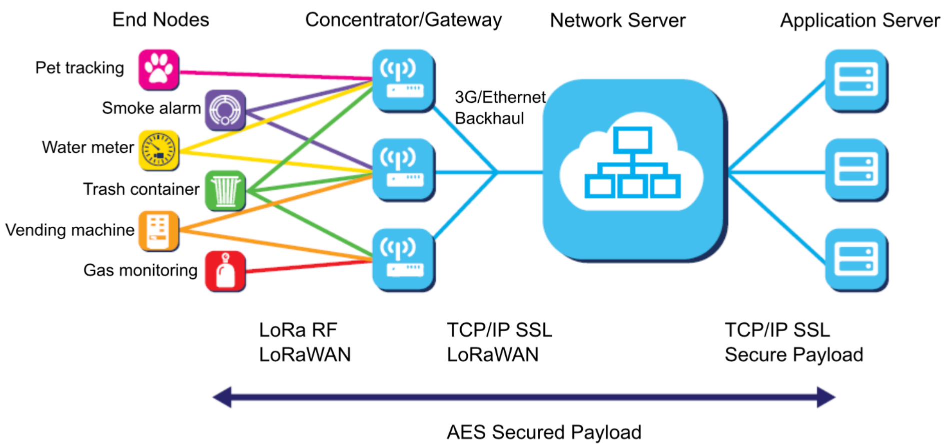


Figure 4.1: Structure of LoRaWAN [19]

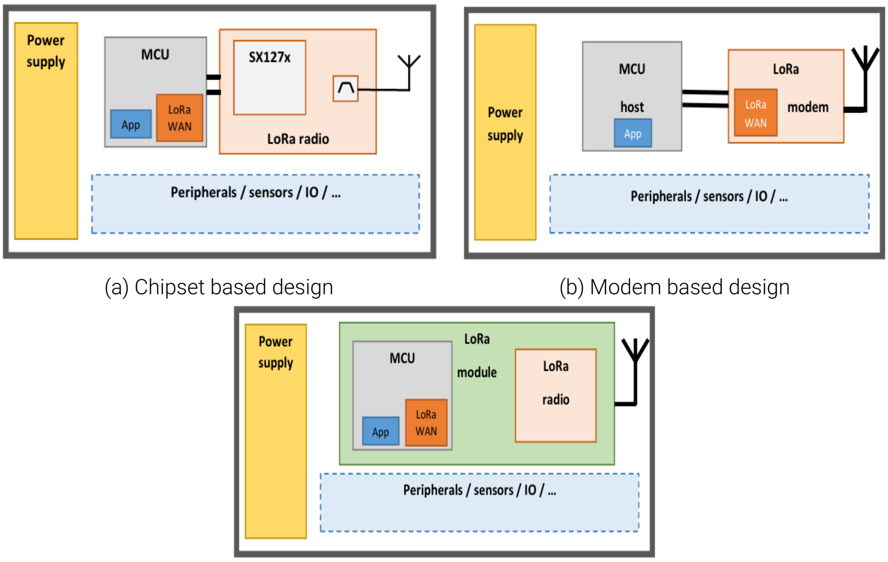
As visualized in Figure 4.1, end-devices are connected with gateways in a star topology. An end-device can connect with multiple gateways. The gateway forwards the message via for example standard IP connections to a network server. The application server is connected to the network server and can consist of various applications depending on user demands.

Public or private network

Because of the popularity of LoRaWAN, large scale networks are already deployed nationwide. These networks are called public networks, as every user can connect to this network when a gateway is nearby. It is also possible to deploy a private network, in which only end-devices can connect with the permission of the network owner. It is reasoned that for this application a private network is more suited. Using a public network usually entails substantial network subscription costs, which can be circumvented by using a private network. Also, public networks may not cover remote areas such as the orchards, a problem that is not present when deploying a private network. Finally, public networks may apply extra restrictions on how many messages an end-device is allowed to send, to prevent the network from congestion. Private network deployment includes the own deployment of gateways and a network server. Fortunately, gateways are widely available and open-source network servers are available as well.

4.2 End-device hardware components

In this section, the hardware components which are needed to create wireless capabilities on the end-device are treated. First of all, to implement the LoRaWAN protocol, LoRa modulation has to be used for sending data. LoRa modulation is a patented modulation technology by Semtech. This means that a LoRa Radio Frequency (RF) transceiver chip, which is manufactured by Semtech, must be used. To further implement LoRaWAN protocol an Microcontroller Unit (MCU) is needed. This can be any MCU as long as it can run the software needed for the LoRaWAN protocol and can communicate with the LoRa RF transceiver chip. Furthermore an antenna is needed. The antenna choice is important for the potential range of the LoRaWAN end-device.
Designing the entire system from scratch is not necessarily needed. Due to the popularity of LoRaWAN, some pre-configured systems already exist. In Figure 4.2, three possible implementations are given.



(a) Chipset based design (b) Modem based design

(c) Module based design

Figure 4.2: Possible configurations for the hardware design [20]

The first option in Figure 4.2a is designed starting with the LoRa RF transceiver chipset from Semtech and an MCU, both chosen by the developer. This means that the LoRaWAN protocol has to be implemented on this MCU. Also, a Printed Circuit Board (PCB) with all the hardware needed to run the LoRa RF chip has to be designed. The second option in Figure 4.2b is to implement a LoRa modem. On this modem, an MCU is implemented with the LoRaWAN protocol, such that this does not have to be implemented on the MCU which is still chosen by the developer. In the last option in Figure 4.2c, the entire LoRa module is already made, only additional application software has to be written.

Implementation choice

First of all, the configuration in Figure 4.2a provides a lot of implementation freedom since the MCU and LoRa RF transceiver chip can be chosen independently. This also gives the opportunity to significantly reduce device cost. From the perspective of power consumption, optimal choices can be made. However, a proof of concept or a prototype with this implementation choice would not be possible due to the time constraints of this project.
The last configuration, where an MCU and LoRa RF transceiver chip are already chosen and implemented, is also not chosen. The cost of these modules is relatively high and this configuration does not allow for much implementation freedom. Furthermore, as another subgroup already focuses on the design and implementation of the MCU it is not desired to incorporate the MCU in the design of the wireless module. The choice is made to use a pre-configured modem as in Figure 4.2b, since this implementation is almost directly usable. Also, large development costs and time are omitted in comparison to the chipset implementation in Figure 4.2a. This still gives freedom in choosing an MCU. However, the extra MCU on the modem, which implements the LoRaWAN protocol, causes a slight increase in power consumption.

4.2.1 LoRaWAN modem

First of all, the LoRaWAN modem must be operating in the European 863 MHz to 870 MHz frequency band since the main focus of this project is in Europe. A comparison is made between three modems implementing LoRaWAN, namely the eRIC-LoRa, iM881A, and RN2483 modem. These modems were chosen as possible candidates based on their availability and documentation compared to other modems not listed here. The modems are compared on their pricing, power consumption in different operation modes, and interface.

Table 4.1: LoRaWAN modem characteristics

	eRIC-LoRa	iM881A	RN2483
Cost	€19.95	€12.09	€10.95
TX / RX current [mA]	- / 15	38 / 11.2	38.9 / 14.2
Sleep current [μA]	15	1.4	1.6
Interface	RS232/UART	I2C/SPI/UART	UART

As can be seen from Table 4.1, the eRIC-LoRa modem is relatively high in cost and also the sleep mode power consumption is significantly higher. Note that the transmission power consumption was not listed in the datasheet. For these reasons, the eRIC-LoRa is omitted from the options. From the two possible options, the RN2483 modem is chosen as the best solution. Although the power consumption of the RN2483 is slightly higher than the iM881A, the lower cost, extensive documentation, and manufacturer availability of the RN2483 in comparison to the iM881A led to this decision. The MCU chosen by the measurement and control group [6] has an UART interface, so communication between the MCU and RN2483 is possible.

4.2.2 Antenna

For the antenna selection, an important choice is made concerning the directivity of the antenna. Certain antennas, such as helical or patch antennas, can have a high directivity in a specific direction, which improves energy efficiency. However, these antennas must be directed to the gateway for optimal performance. For this application, the choice is made to implement an omnidirectional antenna, which transmits equally for a constant azimuth angle. This design choice is made to improve the robustness and deployment of the system. Also, a directed antenna is not suited if multiple gateways are used since the antenna can then only be directed towards one gateway.
Very common omnidirectional antennas are the whip and dipole antenna, which have a donut-shaped radiation pattern. These antennas are considered as an option since along their axis no radiation is present. This is desired since for this application it is already known that no radiation into the air above the device is required. This results in a higher gain into the main lobe of the radiation pattern which is directed to the gateway. In comparison with ceramic chip antennas, the gain of the whip or dipole antenna is higher because the ceramic chip antenna does radiate into the air above the device.

VSWR

The Voltage Standing Wave Ratio (VSWR) is a measure of how well the antenna is matched to a source impedance, which typically is 50 Ohms. The VSWR is the ratio between the maximum and minimum standing wave amplitude, created by interference of a forward and backward traveling wave in a transmission line, given by Equation 4.1:

$$VSWR = \frac{A_{max}}{A_{min}} \tag{4.1}$$

The backward traveling wave is caused by reflections. If the VSWR is equal to 1, no reflected power is present and no standing wave is created. Antennas can be matched perfectly on a specific frequency, but for a larger frequency range small mismatches can exist, resulting in a VSWR larger than 1. When selecting an antenna, it is thus important to consider the largest VSWR in the frequency range of interest and select an antenna with a VSWR close to 1.

Antenna gain

The gain of an antenna is usually expressed in dBi, decibels-isotropic. This is a measure of power gain relative to the theoretical isotropic antenna. As explained later in Section 5.4, the maximum Effective Radiated Power (ERP) of a LoRaWAN end-device is 14 dBm. The ERP is a measure concerning the radiation power of a half-wave dipole antenna. Since the half-wave dipole antenna has a gain of 2.15 dB relative to an isotropic antenna, the maximum Effective Isotropic Radiated Power (EIRP) is 16.15 dBm. There is thus still space for an antenna with a gain of 2.15 dBi to stay under the ERP limit of 14 dBm.

Antenna selection

The antenna should be operating in the 863 MHz to 870 MHz frequency range. The most suited antenna for this application is the whip antenna since there is very low availability of dipole antennas for this application. However, whip antennas require a ground plane for optimal performance and thus it is difficult to estimate the real performance of these antennas. In Table 4.2, three widely available whip antennas are compared based on their pricing, gain, and VSWR. The prices are based on bulk prices found at Digikey, Mouser, Farnell, and RS.

Table 4.2: Whip antenna characteristics

	ANT-868-PW-QW	ANT-8WHIP3H-SMA	ANT-868-OC-LG-SMA
Cost	€6.93	€4.21	€5.60
Peak gain [dBi]	1.6	3.0	0.4
VSWR	1.9	1.4	1.9

The ANT-8WHIP3H-SMA in Table 4.2 is chosen based on the relatively low cost, high gain, and low VSWR. It is however important to test the antenna performance in reality and it is advisable to evaluate the received signal strength at several distances.

4.3 Gateway

Choosing a gateway depends on several factors. First, it is important to specify the operating environment. The gateway needs to be specified to operate in an in- or outdoor environment, with proper operating temperatures. Furthermore, the connectivity to the network server should be considered. This can be accomplished via either Ethernet, WiFi, or cellular networks or a combination of the three. With certain gateways, it is also possible to have a network server running on the gateway directly.
Now, the capacity of the gateway is reviewed. As will be explained later in Section 5.1, LoRa messages are orthogonal when using different Spreading Factors (SFs) in the range of SF7 to SF12. This means that messages on the same frequency channel with a different SF will not interfere with each other. There are relatively cheap single frequency channel gateways available, which only support reception of messages from one frequency channel and one SF simultaneously. Fortunately, also gateways are available supporting up to eight frequency channels. These channels can receive different SF simultaneously in range SF7 to SF12 and thus have significantly higher capacity. However, it must be noted that these gateways only have eight physical message demodulators on board. Thus, only eight packets can be demodulated simultaneously [21]. To increase the network capacity, an 8-channel gateway is chosen. Furthermore, it is important to note that a gateway cannot transmit and receive simultaneously. This means all incoming packets will be blocked when a gateway is transmitting a message to an end-device.

4.4 Network server

The network server ensures correct communication between the end-devices and other applications that handle the data. All the packets which are transmitted on the network are sent to this network server. A function of the network server is discarding the duplicate messages received by multiple gateways on the same network. The network server also handles authentication of the end-devices. Also, it schedules packet transmissions from the gateway to the end-device. Creating a network server would not be needed as some open source options are available. An example of this is the open-source ChirpStack network server [22].

4.5 Application server

To complete the LoRaWAN, an application server is needed. This is the brain of the network that decryptes and encryptes data and handles the network join requests of end-devices. Apart from this, it contains a web Application Programming Interface (API) from which other applications can get the data from the network. ChirpStack offers an open source application server, which also contains a web interface where it is possible to manage users, applications, and end-devices.
Apart from the network, to deliver a complete product, additional applications have to be created which handle the possible actions taken based on the incoming data. When the system is used for research purposes, a front-end application should be made which takes in all the data and creates a 3D temperature map from this data and displays it in a web or desktop interface. When the system is used as a frost prevention system, a back-end application should be running which can send information to a front-end application, such as a mobile or web-app, through which the farmer can be notified of frost with a notification from the app, an email or text message. Also, in this app, the farmer can enter in what period he would like to do measurements to prevent the crops from frost and adjust the critical temperature for specific crops, as listed in the requirements. The back-end application can then also be used to initiate an action depending on the available frost prevention method. These applications all communicate through the application server API.

Chapter 5

LoRaWAN end-devices

In the previous chapter, the hardware considerations were listed to implement the entire Wireless Sensor Network (WSN). In this chapter, first the parameters that can be configured in a LoRaWAN end-device are treated. Then, the LoRaWAN specifications for an end-device are elaborated. Furthermore, the structure of the uplink packets, that is from end-device to gateway, and downlink packets, that is from gateway to end-device, are treated. Next, a maximum communication distance estimation of the end-devices is made. Knowing the packet lengths and maximum communication distance, an energy consumption estimation is made based on different scenarios. Finally, the join procedure and the use of confirmed messages in LoRaWAN are discussed.

5.1 End-device parameter configuration

To create an end-device that can transmit messages, some basic parameters not specified in the LoRaWAN specification, should be configured by the developer of the network. These parameters define how LoRa modulation is performed and can be different for every packet. Each end-device can use a certain Data Rate (DR) to send a message on a certain frequency channel. The DR is composed of the used type of modulation, the spreading factor, and bandwidth. According to a combination of these factors, also a maximum application payload is applicable. In Appendix B.2, different DRs ranging from DR0 to DR7 are listed for devices deployed in Europe. Now, the adaptable parameters in LoRaWAN are explained in more detail.

Center frequency

The center frequency is the carrier frequency on which the message is sent. The carrier frequency which can be used is dependent on the regional regulations. In Appendix B.1 the possible carrier frequencies for Europe and their regulations are specified.

Spreading Factor

The Spreading Factor (SF) stands for the number of raw bits that are encoded in a single symbol. This means that the amount of values encoded in a single symbol is 2^{SF} coming from 2^{bits}. The symbols are encoded in chirps, which are simply a linear frequency sweep across a frequency band defined by the bandwidth and the carrier frequency. This type of modulation is called Chirp Spread Spectrum (CSS). By using the entire channel bandwidth for the chirps, LoRa modulation is very robust to channel noise. To allow for symbols with higher spreading factors to contain more bits, these symbols are encoded on chirps which take twice as long to sweep across the same frequency range. This means that the time to send one symbol doubles for every spreading factor increase. In LoRaWAN, messages with different SFs are orthogonal and cannot interfere with each other. The larger the SF, the more noise resistant the signal is so the maximum distance at which the signal can be received also increases.

Bandwidth

The Bandwidth (BW) denotes the frequency range used by an up- or down chirp. The BW also represents the chirp rate. Thus a higher bandwidth gives rise to a higher data rate. The BW is adaptable with the choice from 125 kHz and 250 kHz for Europe. For the sake of regularity, in this design a bandwidth of 125 kHz is used since DR0 to DR5 use this bandwidth, as listed in Appendix B.2.

Coding Rate

To improve link quality, LoRa modems employ cyclic error coding to perform forward error detection and correction [23]. Depending on the amount of interference on a channel, the Coding Rate (CR) can be changed. The CR is given by $CR = \frac{4}{4+n}$ with $n = 1, 2, 3, 4$. This expression can be interpreted as four bits which are encoded by 5, 6, 7 or 8 transmission bits. The cyclic error coding thus introduces an amount of overhead depending on the robustness to interference that is desired. For this application in a rural area, the lowest $CR = \frac{4}{5}$ is used.

5.2 LoRaWAN specification

In this section, the specifications of LoRaWAN are treated, the different LoRaWAN classes are reviewed and final end-device implementation details are provided.

5.2.1 LoRaWAN network layers

The networking part of an end-device can be divided into three layers depicted in Figure 5.1 [24]. The first layer is the application layer. Here, the data is gathered from the sensors and processed. The application layer passes the data to the Media Access Control (MAC) layer when it needs to be sent. The MAC layer contains all the information and parameters needed to communicate in the network. The MAC layer adds additional headers to the data which are needed for the communication between the network server and end-device. If everything is configured correctly, the MAC layer passes the data to the Physical Layer (PHY) which sends the data using LoRa modulation to a gateway.

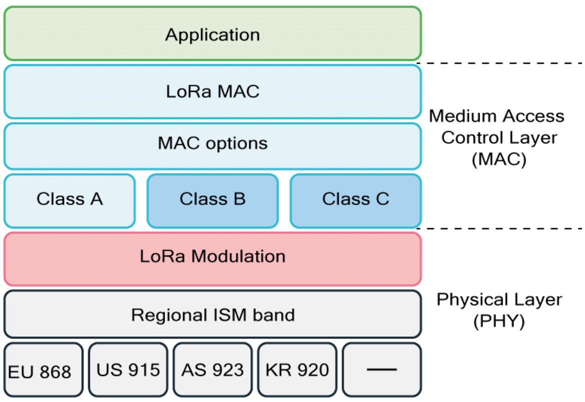


Figure 5.1: LoRaWAN network layers in a end-device [24]

5.2.2 LoRaWAN classes

The LoRaWAN specification describes three class types: A, B, and C. These classes indicate options for the MAC layer in end-devices. Class A is the base implementation of the LoRaWAN MAC layer. This means all end-devices must have the functionality described in the class A specification. Class A has two time slots that are opened after a transmission in which downlink messages can be received. Class B adds the possibility to send data to the end-devices in additional time slots compared to the ones given in class A. This is achieved by having the gateway sending a beacon signal regularly. This beacon is used for timing reference. Finally, in class C the end-device is constantly listening for possible downlink messages. Class A is by far the most power-efficient solution, namely 6 and 2000 times more power-efficient than class B and C, respectively [25]. Thus class A is used for this application.

Class A

Class A specifies that after every transmission of an end-device, two short receive windows need to be opened by the end-device in which downlink messages can be received. In the first window, RX1, the end-device listens for a message on the same Spreading Factor (SF) and frequency as the previous uplink message. The second window, RX2, is used to receive information with a predetermined SF and frequency. The receive slots open after a certain delay specified by RECEIVE_DELAY1 and RECEIVE_DELAY2 as visualized in Figure 5.2.

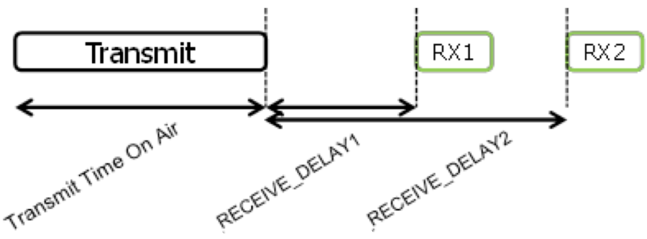


Figure 5.2: Class A end-device receive slot timing [23]

For the RN2483, the RECEIVE_DELAY1 can be adapted by the developer, but a delay of 1000 ms is default. RECEIVE_DELAY2 is fixed to a value of RECEIVE_DELAY1 + 1000 ms in the RN2483. The receive window size of RX1 and RX2 can be set between 0 to 65535 symbols [26]. These window sizes should be large enough to at least detect 5 symbols of a downlink message preamble [23] [25]. The symbol time T_{sym} of a LoRa packet with a certain SF and BW of 125 kHz is given in Equation 5.1 [27] and can be found in Table 5.1.

$$T_{sym} = \frac{2^{SF}}{BW} \tag{5.1}$$

A receive window size of 5 symbols would demand extremely good timing for downlink message scheduling and sending by the network server and gateway, respectively. Therefore, a minimum extra time margin of 10 ms is given. Note that for higher SF, this margin becomes larger since the time of sending one symbol is larger than the given margin. The final receive window sizes corresponding to a certain SF are listed in Table 5.1 below.

Table 5.1: RX1 and RX2 receive window sizes for different SFs

Spreading factor	SF7	SF8	SF9	SF10	SF11	SF12
Symbol time [ms]	1.024	2.048	4.096	8.192	16.384	32.768
Receive window sizes [ms]	15.36	20.48	32.77	57.34	98.30	196.61

The disadvantage of using RX1 is that the downlink message might collide with an uplink message sent on the same channel since the same SF and frequency are required to be used in RX1. The RX2 settings can be set on any particular frequency channel and SF. The default and most useful frequency channel is the 869.525 MHz channel. This channel can only be used for downlink messages and has a more relaxed regulation. The duty cycle limit is 10% and the maximum sending power is 500 mW (27 dBm) [23]. Furthermore, because no uplink messages are sent on this channel, there is no possibility that downlink messages collide with uplink messages. For this reason, it is decided that RX2 primarily is used to return downlink messages. However, the RX1 still might be used in cases it is known that no other end-device will transmit an uplink message. This will be explained later in Section 6.2.

5.3 Up- and downlink packets

The RN2483 modem has a hexadecimal interface. As the payload of a LoRaWAN message is per byte, two hexadecimal values are used to represent each byte sent. To make use of the payload efficiently, the data is translated from binary format to hexadecimal. To easily distinguish between different parts of the data and translate the binary data into decimals again in the MCU, the size of all different parts of the data is determined independently. There has been made a distinction between measurement periods, namely when temperature measurements are required to protect the crops or not. If this is not the case, this is referred to as the off-season period.

An uplink message will always contain the following payload:

- Energy level indicator: 1 byte
- Measured temperatures of 5 end-devices: 10 bytes

The energy level indicator range is between 1% to 100%. This range is chosen since if the end-device has a zero energy level, the uplink message cannot be sent. Furthermore, the maximum temperature range of the temperature sensor chosen by the measurement and control group [6] is between -40 °C and +125 °C with an accuracy of one decimal place. Although the practical temperature range will be smaller, still 2 bytes are required to represent these numbers using two's complement representation. Finally, the temperature measurements of all five end-devices and the energy level indicator will be concatenated to a static packet structure. The total payload size of an uplink message is thus equal to 11 bytes. As described by the measurement and control group [6] and taking into account the maximum application payload as displayed in Table B.3, every four hours a message is sent with hourly temperature measurement data in the off-season period.

A downlink message may contain the following information:

- The real time: 4 bytes
- Assigned time slot: 3 bytes
- Critical temperature: 2 bytes
- Off-season period: 1 byte

The exact reason for the specific types of data included in a downlink message is explained in Chapter 6. The first part of the downlink message is the daily real time, which is sent with a millisecond resolution. To present the time of the day with a millisecond resolution, 26.4 bits are required, which results in a 4-byte payload, as calculated below.

$$Required\ bits = log_2(24 \cdot 60 \cdot 60 \cdot 1000) \approx 26.4\ bits$$

Furthermore, the time slot is assigned in a time frame of 5 minutes, as the maximum transmission rate of an end-device is once every 5 minutes, as listed in the requirements. Thus, 18.2 bits are required, resulting in 3 bytes, as calculated below

$$Required\ bits = log_2(5 \cdot 60 \cdot 1000) \approx 18.2\ bits$$

The critical temperature is represented using 2 bytes, following the explanation in the uplink message structure. To indicate whether measurements are required to protect crops or if the off-season period is active, a yes "Y" or no "N" ASCII symbol, respectively, is sent when a change in mode is requested.
The downlink message has a worst-case message size of 10 bytes. Note that the size of this message is not necessarily the same, as less information might be required to be sent in certain cases.

5.4 Duty cycle limitation

LoRaWAN messages of end-devices are limited by duty cycle regulations. This means that an end-device has a minimum off period T_s, which is dependent on the duty cycle d and the Time On Air (TOA) of a transmission T_a, given by Equation 5.2:

$$T_s = T_a(\frac{1}{d} - 1) \tag{5.2}$$

The TOA of a LoRa transmission can be calculated as described in Appendix B.3. In Appendix B.1, the ETSI and LoRaWAN frequency band regulations are given. It can be seen that three frequency bands, namely 865.0 MHz - 868.0 MHz, 868.0 MHz - 868.6 MHz, and 869.7 MHz - 870.0 MHz allow for a maximum effective radiated power of 25 mW (14 dBm) and a maximum duty cycle d of 1%. According to the LoRaWAN specification, the first three channels located on 868.1 MHz, 868.3 MHz, and 868.5 MHz with DR0 until DR5, which are listed in Section B.2, must be implemented on every end-device. Also, the gateway should listen on at least these three channels. Earlier it was concluded that an 8-channel gateway is used. Thus, also other frequencies in the above-specified frequency bands are implemented on devices which are: 867.1, 867.3, 867.5, 867.7, and 869.8. These all have a maximum duty cycle d of 1% and maximum effective radiated power of 25 mW.

5.5 Maximum communication distance

In this section, the maximum attainable distance in a rural environment is investigated. This is mainly dependent on the transmission power P_{TX} and the receiver sensitivity S_R. The transmission power can be programmed on the RN2483 chips in a range of -4 to 14.1 dBm in 19 steps. Gateways have a higher sensitivity than end-devices. The main bottleneck of this system is thus sending a message from gateway to end-device. In the downlink receive window RX2, the gateway is allowed to use a maximum power of 27 dBm in the chosen 869.40 MHz to 869.65 MHz channel, which does not introduce any bottlenecks. The main bottleneck is sending downlink messages in RX1, in which the gateway is obliged to use the same channel and spreading factor as the uplink message. The Effective Radiated Power (ERP) restriction of 14 dBm also applies to the gateway in RX1. Thus, the sensitivity corresponding to a certain SF of the RN2483 is given in Table 5.2 to calculate the maximum communication distance.

Table 5.2: Sensitivity of a LoRa end-device

Spreading Factor	SF7	SF8	SF9	SF10	SF11	SF12
Sensitivity [dBm]	-125	-128	-131	-134	-136	-137

To find the maximum distance d, the maximum allowed path loss L_{path} in dB is found from rewriting the link budget in Equation 5.3:

$$L_{path}[dB] = P_{TX} + G_{TX} + G_{RX} - L_{cable\_TX} - L_{cable\_RX} - S_R - L_{margin} \tag{5.3}$$

Where G_{TX}, G_{RX}, L_{cable_TX} and L_{cable_RX} account for the antenna gain of the transmitter and receiver, and cable loss of the transmitter and receiver, respectively. The link margin L_{margin} accounts for a margin between the minimum expected received power and the sensitivity of the receiver.
To calculate the maximum distance, the free-space path loss model is used with path loss exponent $\alpha = 3$, which in general is on the border of rural to urban area. The path loss is given by Equation 5.4 [28] and can be rewritten to obtain the maximum distance in Equation 5.5.

$$L_{path} = \left(\frac{4 \cdot \pi \cdot f}{c}\right)^2 \cdot d^{\alpha} \tag{5.4}$$

$$d = \left(L_{path}\left(\frac{c}{4 \cdot \pi \cdot f}\right)^2\right)^{\frac{1}{\alpha}} \tag{5.5}$$

Where c is the speed of light in vacuum and f is the carrier frequency of 868 MHz. The free space path loss model in Equation 5.4 and 5.5 is used as an upper bound for the maximum distance with the path loss exponent near urban environments. Finally, the maximum distances of different SFs versus transmission powers are obtained in Figure 5.3 below.

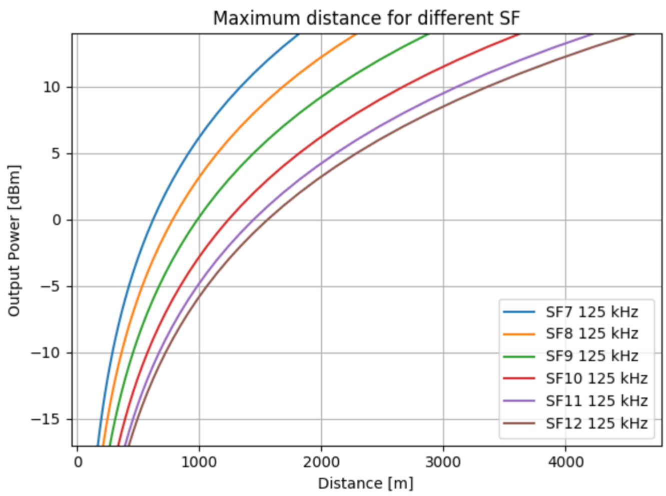


Figure 5.3: Maximum distance obtained with different SFs and transmission power.

In Figure 5.3, a link margin L_{margin} of 10 dB is introduced to provide a safety margin such that no messages are lost due to approaching the maximum sensitivity of the receiver. Also, the peak antenna gain of 3 dBi for the ANT-8WHIP3H-SMA, is not embedded into the calculation but transmit and receive antenna gains of 0 dBi are used instead to account for impedance mismatching and non-optimal performance of the antenna. Figure 5.3 thus shows a worst-case scenario for the maximum attainable distance for a certain transmission power and SF.

According to the requirements, a maximum distance of 1 km must be obtained in a worst-case scenario. Important to note is that the output power setting on the RN2483 is limited by the antenna gain, as explained in Section 4.2.2. A maximum E.R.P of 13.15 dBm is now allowed for the RN2483. On the RN2483, the power can be adjusted in steps, and thus the maximum output power should be set at 12.5 dBm [29]. From Figure 5.3 it can be said that this is not a problem for all SFs. Also, the output power of end-devices with a higher SFs can be lowered to reduce power consumption. To give some distance margin such that a worst-case scenario is not exactly at 1 km, all output powers corresponding to a specific SF are adjusted to a power setting in the RN2483 which accommodates a maximum distance of 1.5 km. These output powers are listed in Table 5.3 in the consecutive section.

5.6 Power breakdown

In this section, the power breakdown of the LoRaWAN modem RN2483 is made. With the knowledge that the device is class A, that the up- and downlink messages have a certain length, and that the distance requirement can be fulfilled, an energy consumption estimation is made. An important factor influencing energy consumption is the Time On Air (TOA) of the up- and downlink messages. The TOA of LoRa packets is calculated according to the formulation given in Appendix B.3. Note that a downlink messages does not have a CRC included, thus downlink TOA is reduced [23]. Now, different scenarios for the energy consumption per transmission are created based on the reception of acknowledgements (ACKs) from the gateway in a certain receive window, RX1 or RX2.

- Scenario 1: Uplink transmission without ACK.
- Scenario 2: Uplink transmission with ACK in RX1.
- Scenario 3: Uplink transmission with ACK in RX2.
- Scenario 4: Uplink transmission with failed ACK in RX1 and successful ACK in RX2.

In the first scenario, no ACKs are sent to the end-device. This means that both receive windows are open for the specified window size in Table 5.1. In the second scenario, the ACK is received in RX1 and RX1 will thus be opened for the entire time on air of the downlink packet. Note that, as in this scenario the downlink packet is received at the correct end-device, the RX2 window will not be opened [23] and thus power is saved. In the third scenario, the ACK is received in RX2 and this window will also be opened for the entire time on air of the downlink packet. In the fourth and worst-case scenario, an ACK which was targeted at another end-device or an ACK with an error is received in RX1. In this case, the message however has to be demodulated and RX1 is opened for the entire time on air of the downlink packet. Now, the end-device also opens RX2 to attempt a successful reception of a downlink message.

In Table 5.3 below, for each SF with a bandwidth of 125 kHz, the uplink and downlink TOA are listed. Also, the RN2483 output power settings are listed according to the datasheet [30] and the transmission currents corresponding to these output powers are listed.

Table 5.3: Parameters for power analysis

Spreading factor	SF7	SF8	SF9	SF10	SF11	SF12
Uplink 11 bytes TOA [ms]	36.1	72.2	144.4	247.8	495.6	991.2
Transmission power [dBm]	12.5	8.1	5.8	2.5	0.4	-0.6
TX current [mA]	36.5	31.2	28.8	24.7	22.3	21.2
Downlink 10 bytes TOA [ms]	36.1	62.0	123.9	247.8	495.6	827.4

In Figure 5.4 below, the consumed energy per uplink transmission for all SFs is given, with a payload of 11 bytes. Also, in Figure 5.5 an energy consumption analysis is made of an SF9 end-device. This analysis includes one uplink transmission of 11 bytes and one downlink reception of 10 bytes in RX2, thus implementing the third energy consumption scenario.

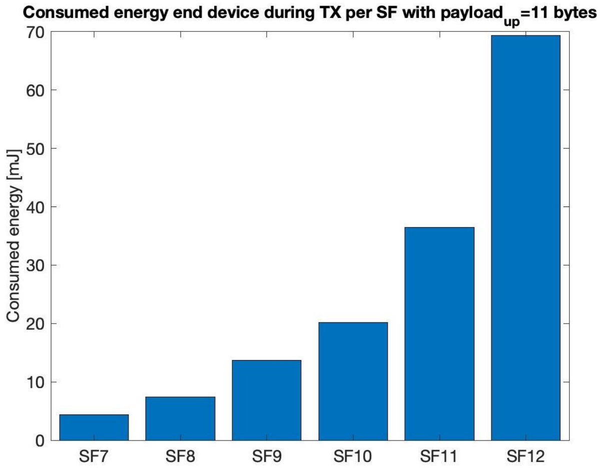


Figure 5.4: Transmission energy per SF for an 11 byte uplink packet

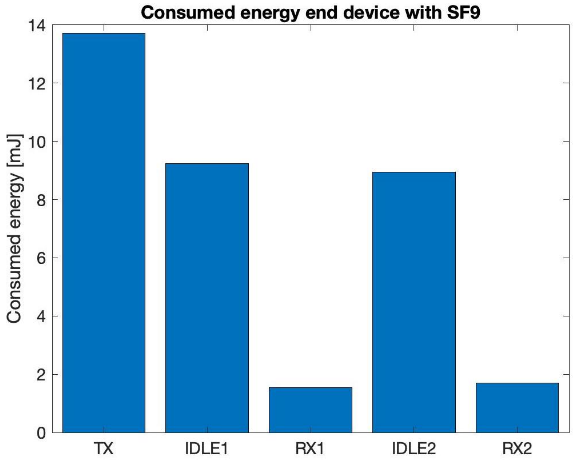


Figure 5.5: Energy consumption of an end-device with transmission on SF9

As explained in Section 5.2.2, the first receive delay equals one second by default. In Figure 5.5 this is referred to as IDLE1. Then, the first receive window RX1 opens and no message is received in this case. Subsequently, the RN2483 is in IDLE2, which has a length equal to one second minus the RX1 window size. Then, RX2 is opened and with successful detection of 5 downlink preamble symbols, the RX2 window remains open for the entire TOA of the downlink packet.

Finally, in Figure 5.6 below, the energy consumption for all four previously described scenarios are plotted for each SF.

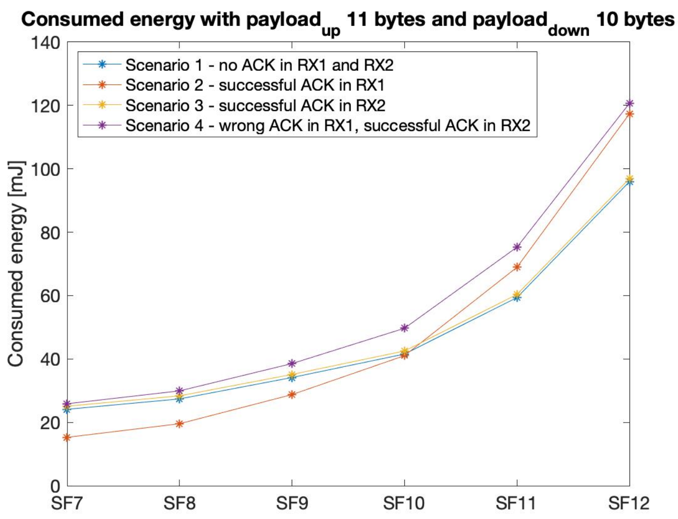


Figure 5.6: Total energy consumption for all SFs under different scenarios

As can be seen in Figure 5.6, the energy consumption increases exponentially with the SF. To minimize energy consumption in an end-device it is decided that SF11 and SF12 are not used. In Figure 5.6 it can be seen that when only using SF7, SF8, SF9, and SF10, the maximum energy used per transmission is 49.69 mJ (13.80 µWh). The remaining time in which the RN2483 is not sending, it is in sleep mode consuming 5.28 µW.

5.7 Network join procedure

In LoRaWAN, two join procedures are defined, namely Over-The-Air Activation (OTAA) and Activation By Personalization (ABP). In OTAA, an end-device joins the network by sending a join request. The network server then responds with a join accept message, which sends all necessary authentication keys, device address, and other settings to the end-device.
In ABP, four session keys and the device address are directly programmed in the end-device. This enables the device to directly join the network with the given parameters and bypass the join-accept procedure as in OTAA. For this project, ABP is used since knowing the device address in advance is useful to determine the position of each end-device. When joining the network using ABP, an end-device transmits the ResetInd MAC command for all uplink messages until it receives a ResetConf command from the network according to the LoRaWAN specification V1.1. However, the firmware of the RN2483 does not support the LoRaWAN specification V1.1 yet. In earlier versions, the end-device can directly send messages to the network, knowing all necessary parameters.

Rejoining the network

In case an end-device runs out of power, the end-device waits until it has enough energy to properly function again [6]. In that case, a rejoining procedure is required since the end-device lost track of time. In LoRaWAN V1.1 [23], the end-device then performs a rejoining request, which again includes sending the ResetInd MAC command and wait for a ResetConf command. This is done using SF11 or SF12 to prevent collisions with other already joined end-devices. After re-initialization, the device is required to use the configuration of when the device was first connected to the network. During this re-initialization, the network again informs the end-device with the time, its allocated time slot, the critical temperature, and whether it the off-season period or not. Furthermore, it is decided that if an end device did not receive any downlink messages for 20 consecutive times, a rejoining procedure is evoked. When the system is in off-season mode it waits 5 consecutive times before rejoining.

5.8 Confirmed messages

In LoRaWAN, there is the possibility to force a confirmed message from the end-device and also from the gateway side.

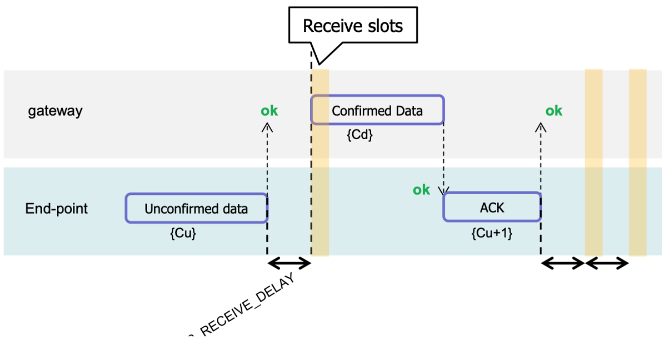


Figure 5.7: Confirmed downlink messaging scheme [23]

Due to the non-optimal downlink capabilities of LoRaWAN class A, no uplink confirmed messages are used. Confirmed downlink messages however will be used, as visualized in Figure 5.7. The gateway sends a downlink message which should be acknowledged by the end-device by setting the ACK bit high [23]. This acknowledgement does not necessarily have to be sent directly as in Figure 5.7, but can be included in the next uplink data message from the end-device. In this way, more reliability is added without extra energy usage.

Chapter 6

Network implementation

In this chapter, the implementation of the LoRaWAN is treated. First, a network in which all 2000 end-devices send at random times is investigated. According to packet loss calculations and downlink messaging capabilities of such a network, a solution that includes packet scheduling using time slots is introduced. As described by the measurement and control group [6], end-devices in this scheduled system transmit once every 5, 10, 15, 20, and 30 minutes based on the current temperature and the amount of energy left in the end-device.

6.1 Packet collisions with random transmissions

To create a network with the least amount of collisions as possible, all eight frequency channels are used. After each transmission, LoRaWAN end-devices are hopping to another frequency of the eight frequency channels, resulting in an average amount of 250 end-devices per channel. When end-devices would randomly send in 5 minutes, one method to reduce the chance of uplink collisions as much as possible is to distribute end-devices over all Spreading Factors (SFs) in such a way the total Time On Air (TOA) of a group of end-devices with the same SF is equal to the other groups with different SFs. The chance of collisions is investigated using this distribution of end-devices. For a packet time p, the probability of k transmissions in this packet time following a Poisson distribution is equal to:

$$P(X = k) = \frac{(\lambda p)^k}{k!} e^{-\lambda p}$$

With λp representing the expected number of transmission packets during a packet time p, where λ is the rate at which arrivals occur. For a successful transmission of a packet on time $t = T$, during the interval $[T - p, T + p]$ no transmissions of other end-devices with the same frequency and SF must occur. The chance of zero other packet transmissions on the same frequency and SF in this time is $P(X = 0) = e^{-2\lambda p}$. Then, this chance is multiplied by the expected number of end-devices within the channel with the same SF. By doing so, the total throughput and thus also the total packet loss of the network can be found when end-devices send at a random moment in 5 minutes. In Figure 6.1 below, the packet loss percentages are given against the payload for different combinations of SFs.

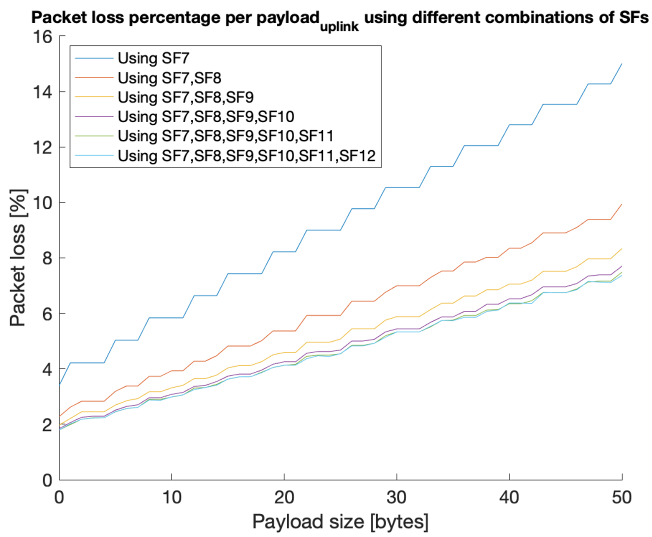


Figure 6.1: Packet loss percentage with 2000 end-devices sending randomly in 5 minutes

It can be seen that with an uplink payload of 11 bytes, the packet loss percentage is about 3% when using four SFs. Note that the different plots show some staircase pattern because the TOA of LoRa packets does not increase linearly with payload but increases in steps of a few bytes. Furthermore, the effect of reducing the amount of SFs becomes more significant when using three or fewer SFs. This is because when using all SFs, relatively fewer end-devices use SF11 and SF12, due to the distribution of end-devices such that the total TOA of all SF groups is equal. Thus, the lower the SF that is not used, the more end-devices have to be redistributed over the remaining SFs, resulting in a higher packet loss.

It is now important to investigate the feasibility of sending acknowledgements to the end-devices. As stated in Section 4.3, a gateway cannot receive and transmit at the same time. So using a single gateway for sending and receiving messages might cause a large loss of uplink messages when downlink messages are sent. To get a good idea of the limitations of using a single gateway the network is simulated, which is discussed in Section 6.3. There, a possible solution to this problem is presented.

6.2 Scheduling system

In the previous section, a significant packet loss rate is calculated. Now, a scheduling system is designed to assure lower or no packet loss.

6.2.1 Time slots

The designed scheduling system assigns a time slot to every end-device in which the end-device is allowed to send. These time slots are distributed over 5 minutes, which is the maximum transmission rate of an end-device. As in LoRaWAN, end-devices are hopping to a different frequency out of eight different implemented frequencies after each transmission, these time slots in 5 minutes are created using all devices with the same SF. The design of the time slots for a certain group of end-devices with the same SF is thus independent of other groups of end-devices with the same SF. In Figure 6.2 below, frequency hopping of two end-devices that have the same SF is visualized. The used carrier frequencies in Figure 6.2 are chosen and used for simulation purposes to utilize all frequency bands with 1% duty cycle efficiently.

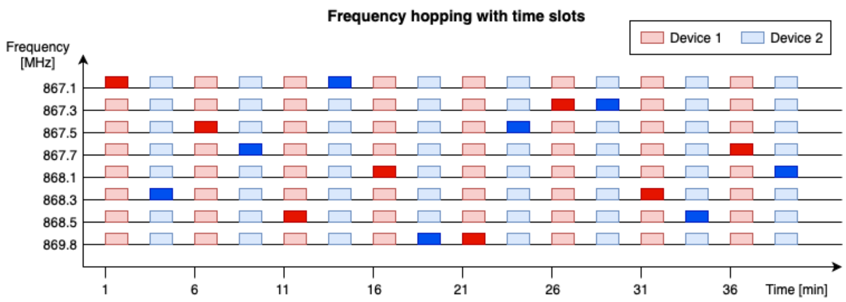


Figure 6.2: Frequency hopping with time slots

When an end-device sends on a certain frequency, the blue or red transmission box is highlighted. Their time slots may not overlap, since that would cause a collision at minutes 21 and 26 for example. When creating a scheduling system in which time slots are used, it is important to investigate the time between two messages of the same SF. This time must be large enough to allow for time drifting of the Microcontroller Unit (MCU) clock [6] to prevent collisions. Also, RX1 downlink messages may not collide with uplink transmissions. By distributing the end-devices over the SFs in such a way RX1 downlink messages do not overlap with uplink transmission slots, more reliable downlink messaging is obtained. Note that if RX1 does overlap with an uplink transmission slot, a collision occurs with a chance of 0.125 since the other end-device must accidentally use the same carrier frequency as the downlink packet. In Appendix B.4, an investigation is made of RX1 feasibility when end-devices are spread over the SFs such that the total TOA is equal. This results in RX1 window and uplink transmission window overlap and thus collisions might occur. Distributing the end-devices according to Table 6.1 results in collision-free RX1 slots, as illustrated in Figure 6.3.

Table 6.1: Parameters for reliable RX1 downlink messages

Spreading Factor (SF)	SF7	SF8	SF9	SF10
Number of end-devices	730	700	400	170
Inter-message uplink time [ms]	374.9	356.4	605.6	1517
Min. RX1 to uplink message time [ms]	161	142	232	269

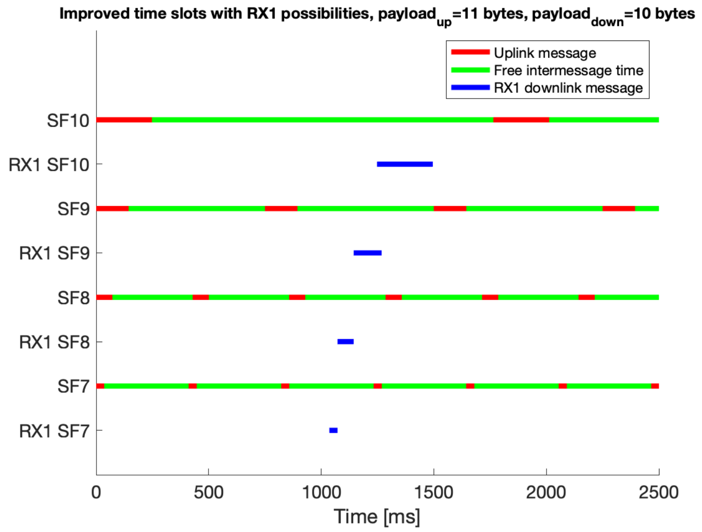


Figure 6.3: Time slots allowing for RX1 downlink messages

6.2.2 Clock drift

In this section, the influence of clock drift of the end-device is discussed. Due to time drifting of the MCU clock, the local recorded time in an end-device changes compared to other end-devices. To prevent a transmission window from overlapping with other transmission windows or the RX1 window, time correction is necessary. The minimum time between transmission windows is the uplink inter-message time of 356.4 ms, corresponding to end-devices with SF8 sending every 5 minutes. This means that end-devices using SF8 should never shift more than 356.4 ms away from their original assigned time slot when transmitting every 5 minutes.

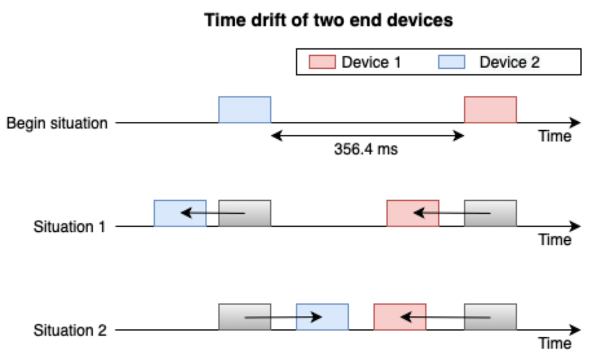


Figure 6.4: Time drift of two end-devices of the same SF. Situation 1, both devices have negative time drift. Situation 2, one device has positive time drift and the other device has negative time drift.

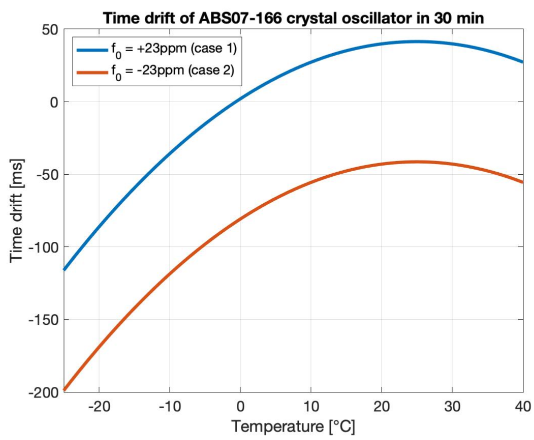


Figure 6.5: Time drift of the crystal oscillator versus temperature

Two possible situations of two time drifting end-devices are depicted in Figure 6.4. Based on the temperature of the end-devices, either situation 1 or 2 is present. In the worst-case scenario, one end-device

follows case 1 and the other end-device follows case 2 of Figure 6.5. In this figure, case 1 and 2 are upper- and lower bounds for the end-device clock time drift, presented by the measurement and control group [6]. In Figure 6.4, the worst-case scenario in situation 1 is when the blue end-device is corrected to its original time slot, depicted in grey and the red end-device has drifted maximally to the left. Therefore, the maximum allowed time drift is equal to the time between two uplink messages. In situation 2, the worst-case scenario is when one end-device has a positive maximum time drift and the other end-device has a maximum negative time drift, as visualized in Figure 6.4. In that case, the maximum allowed time drift is smaller since the time slots of both end-devices are drifting towards each other.
The two end-devices do not necessarily have the same temperature. Thus, a relative temperature difference between end-devices of 5 °C is taken into account. Temperature change caused by the use of wind machines results in approximately 3 °C [31]. To provide some margin for temperature differences across the orchard, a 5 °C temperature difference between two end-devices is used for this worst-case scenario.

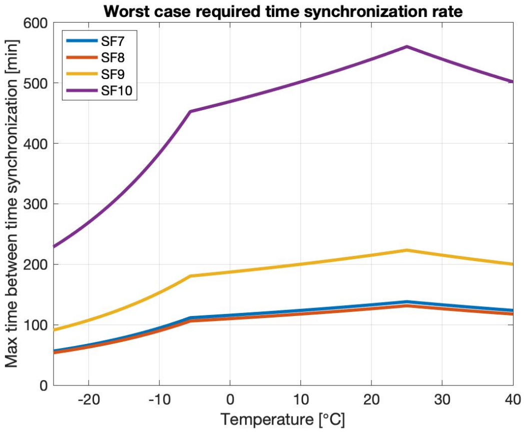


Figure 6.6: Maximum time between synchronisation messages to prevent transmission window overlap

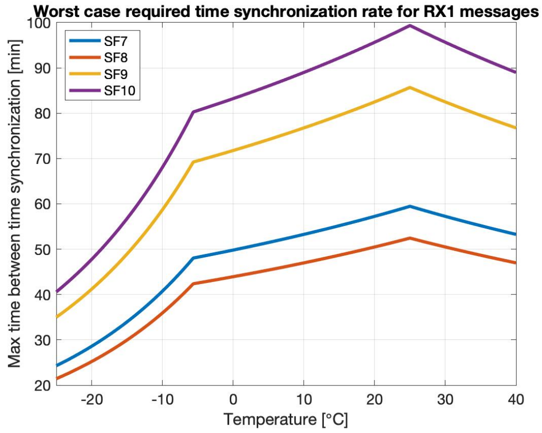


Figure 6.7: Maximum time between synchronisation to prevent RX1 and uplink transmission window overlap

In Figure 6.6, the maximum allowed time between time synchronizations for a transmission rate of 5 minutes is plotted. In Figure 6.6, it is shown that SF8 requires the most frequent time synchronization, which is approximately once every 2 hours for 0 °C. Concerning RX1 transmissions, from Table 6.1, it can be concluded that 142 ms at SF8 is the smallest time between an RX1 window and an uplink transmission window. In Figure 6.7, the maximum time between time synchronizations is plotted as a function of temperature for the four SFs. It can be seen that RX1 downlink transmissions might collide with SF8 uplink transmissions after approximately 43 minutes without time synchronization at a temperature of 0 °C. For other SF, the required time between time synchronizations is larger since the RX1 window to uplink transmission window time is larger. Thus, when sending every 5 minutes, a time synchronization message is necessary at least every 43 minutes when implementing RX1 downlink messages.

6.2.3 Packet processing delays

Finally, the inaccuracies introduced by processing delays and different packet Time On Air (TOA) with different payload sizes are investigated. For each group of end-devices with the same SF, processing delays in the gateway for generating a packet and processing delays in the end-device for demodulating the downlink message are considered constant. A constant processing delay causes a slightly wrong time synchronization of the end-device. This however does not introduce difficulties as simply all time slots of the group of end-devices with the same SF are shifted in time.

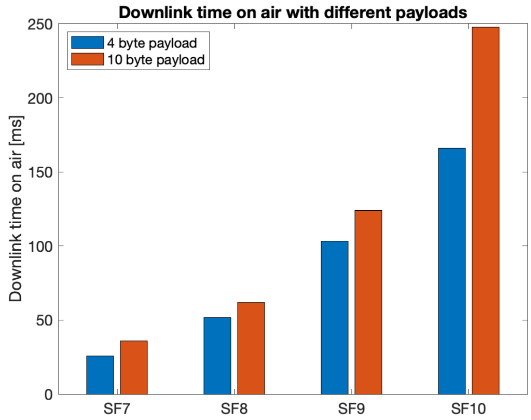


Figure 6.8: TOA of downlink messages with payload sizes of 4 and 10 bytes

Sending a time synchronization message with different payload sizes and thus a different message TOA causes significant time synchronization differences, as illustrated in Figure 6.8. A time synchronization message has a minimal size of 4 bytes. However, a full downlink message contains 10 bytes. These time synchronization issues can be solved by adding the downlink packet TOA as an offset to the synchronization time sent.

6.3 Simulation

To support and verify the decisions made in Sections 6.1 and 6.2, the network is simulated. The simulation is a discrete-time simulation using the Simpy library in Python [1]. This means that the simulation is not a continuous-time simulation but rather an event-based simulation. The created simulation is based on another Simpy LoRaWAN simulation [32]. First, a network simulation is performed which checks single gateway performance. Then, network simulations are performed regarding collisions of random transmissions to verify the previously conducted collision calculations. Furthermore, simulations of the scheduling system are done regarding addressing end-devices with downlink messages.

6.3.1 Simulation structure and properties

In Figure 6.9 below, the structure of the simulation program is illustrated. The most important sub-parts of the simulation program are briefly discussed.

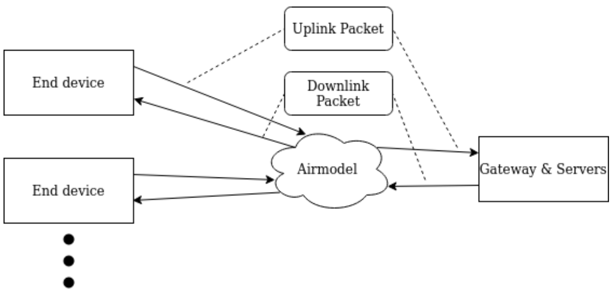


Figure 6.9: The structure of the simulation program

End-devices

In the simulation, the end-devices are all separate instances of an end-device class that contains all the information that an end-device in the orchard would contain as well, such as its location, spreading factor and assigned time slot. The end-device creates an uplink message and a message sending event. After

[1]The code can be requested as it is now stored in a private repository

that, the end-device creates message receive events that open the receive windows. If the entire send and receive cycle is completed, the end-device goes back to sleep mode until it reaches the next time slot.

Air interface

The air interface is used to determine whether different packets collide when they are being sent. The air interface is also used to calculate the received signal strength of packets at the receiving side, according to the path loss model described in Section 5.5.

Gateway and servers

The gateway and servers are implemented as one entity. This entity simply checks if uplink messages are received and downlink messages are scheduled and sent if this is possible according to the duty cycle limitations.

6.3.2 Results

Single gateway

The first simulation checks the system performance when only one gateway is used. In this simulation, 2000 end-devices randomly send an uplink message to the gateway once every 5 minutes and the gateway tries to send a downlink message to all end-devices once. In the simulation, the gateway starts with sending all end-devices a downlink message once. This is done in a best-effort way in which the gateway sends a downlink message when an end-device is ready to receive in either RX1 or RX2 and the gateway is not restricted by the duty cycle on the RX1 or RX2 carrier frequencies. As can be seen in Figure 6.10, in a period of half an hour a total of 1600 messages will be lost, which is about 13% of the total messages sent. A solution to this problem is using an additional single channel gateway dedicated to transmitting downlink messages.

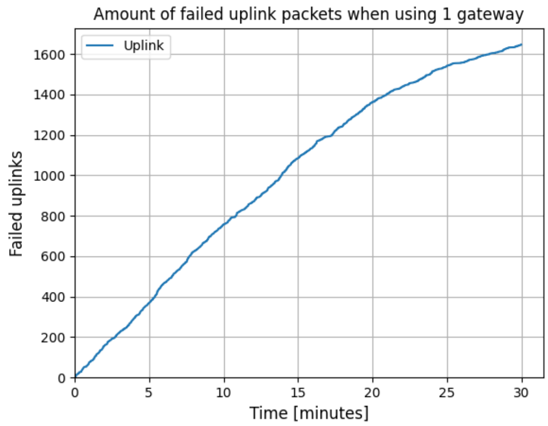


Figure 6.10: Amount of blocked and collided uplink messages using one gateway

Random transmissions

The next simulation done verifies the decision to use time slots instead of sending the messages randomly. Thus a simulation is performed in which each end-device sends at a random time every 5 minutes and the end-devices are addressed with a downlink message once. One gateway is used for receiving messages and another gateway is used for sending downlink messages. In Figure 6.11, the amount of collisions is plotted. In a span of half an hour, 320 uplink collisions happened. That is 2.7% of the total 12000 uplink packets sent. For the downlink messages, 6 collisions were detected. During the simulation 252 downlink messages were sent via receive window RX1, which resulted in 6 collisions and thus the collision rate is around 2.4%. This validates the earlier calculated collision percentage of approximately 3% and supports the idea of using time slots for more reliability. In Figure 6.12 the addressing time of end-devices that send at a random time every 5 minutes is given. From this figure, it can be seen that addressing 2000 randomly sending end-devices takes approximately 35 minutes.

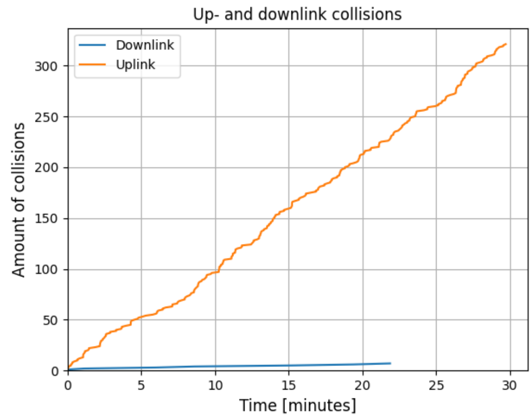


Figure 6.11: Amount of up- and downlink collisions using end-devices transmitting randomly in 5 minutes

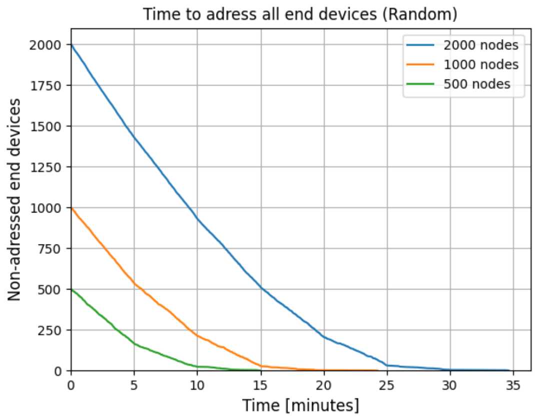


Figure 6.12: Downlink addressing time of end-devices transmitting randomly in 5 minutes

Downlink throughput

Also, the downlink throughput of the system is simulated when using scheduling. This is an important metric, as it decides how often end-devices can receive a time synchronization message such that time drift of end-devices is counteracted. In Figure 6.13, it is shown that the total time to send a downlink message to all 2000 end-devices, sending every 5 minutes in assigned time slots, is about 27 minutes. Also, in Figure 6.13 simulation results for 1000 and 500 scheduled end-devices are shown, resulting in an addressing time of 17.5 and 14 minutes, respectively.

Also, a simulation is done for addressing all 2000 end-devices at different end-device transmission rates, which is depicted in Figure 6.14. The large difference in time can be explained by the fact that when end-devices send less frequently the gateway also has fewer possibilities to send a downlink message.

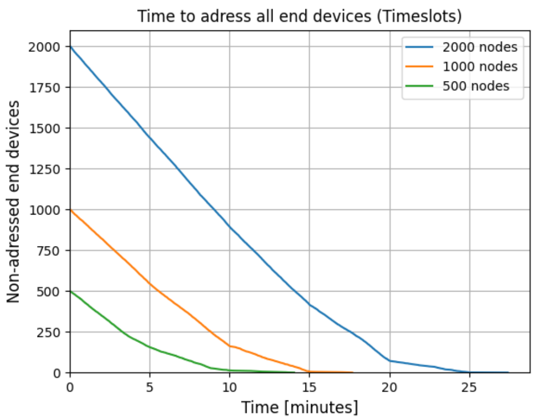


Figure 6.13: Downlink throughput when using time slots

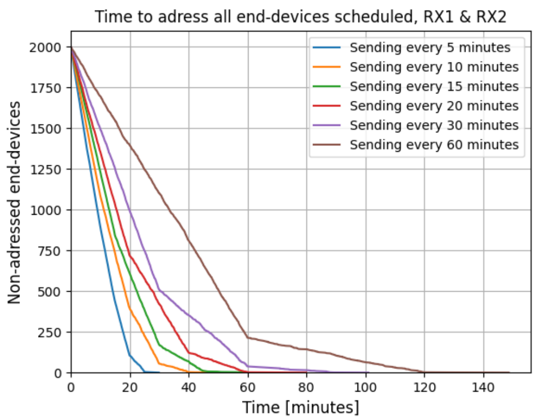


Figure 6.14: Downlink throughput for different transmission rates

Time synchronization messages

Finally, it is important to investigate the feasibility of using time slots concerning end-device clock time drift. In case the critical temperature is set at 0 °C, the transmission rates of end-devices can change on a relatively small temperature scale. As treated by the measurement and control group [6], a transmission rate of once every 5, 10, 15, and 20 minutes occurs at temperatures 1.38, 2.08, 2.77, 4.15 °C respectively. For all temperatures above 4.15 °C, once every 30 minutes a transmission is performed.

Thus, at temperatures below 1.38 °C, the end-device is transmitting every 5 minutes. From Figure 6.6, it can be seen that at this temperature a time synchronization message is required approximately every 2 hours. At lower temperatures of -10 °C, a time synchronization message is required every 90 minutes to prevent transmission window overlap. From Figure 6.7, it can be seen that at -10 °C a time synchroniza-

tion message is required every 35 minutes to prevent RX1 and uplink transmission window overlap. For end-devices transmitting every 5 minutes, the required time synchronization rate to prevent transmission window collisions can be obtained, as from Figure 6.13 it becomes clear that almost all end-devices can be synchronized in a cycle of approximately 27 minutes. Then, two cycles take approximately 54 minutes, which is the maximum time between two synchronization messages for an end-device. This is far below the required 90 minutes at -10 °C and collisions can even be prevented at -25 °C. However, RX1 window and uplink transmission window overlap cannot be prevented. In a non-ideal situation, the time between two synchronization messages of an end-device can be approximately 54 minutes. From Figure 6.7 it can be seen that the required synchronization time is below 54 minutes for devices with SF8.
Since the use of RX1 appears to be difficult to realize, although collision chances are very small, an investigation is done of using RX2 for addressing end-devices with downlink messages only. In Figure 6.15, in which only RX2 is used, the time it takes to send all end-devices downlink messages is found. A slightly worse result is found in comparison to using both RX1 and RX2 in Figure 6.14. The use of RX1 appears to have a small impact on performance improvement.

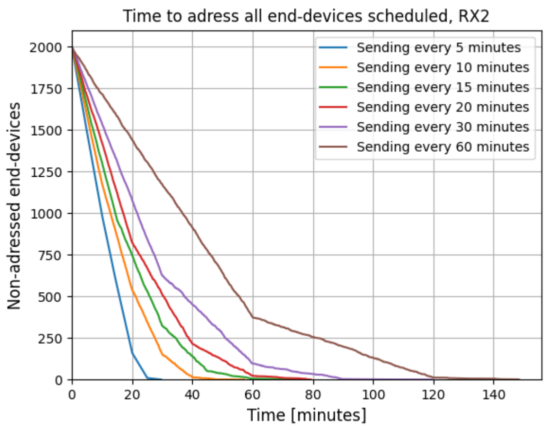


Figure 6.15: Time to adress all end-devices for transmission rates using only RX2

Using the RX2 receive window only, it is possible to prevent transmission window overlap for transmission rates of 5, 10, 15, and 20 minutes, as within the required synchronization period of two hours, almost all end-devices can be addressed with a downlink message in two downlink cycles. For a transmission rate of once every 30 minutes, it can be seen in Figure 6.15 that after one hour, approximately 100 end-devices are still non-addressed. To solve this problem, the second downlink addressing cycle should be started earlier, prioritizing earlier non-addressed end-devices. In this way, a network with scheduled end-devices is created with the prevention of collisions.

It must be noted that all the above-described observations are based on Figure 6.7, which shows a worst-case time between synchronizations for a very crowded scheduling scheme based on all end-devices sending every 5 minutes. At a lower transmission rate, fewer time slots are used, and thus also the required synchronization times become larger. However, it must be noted that all end-devices individually decide which transmission rate should be used. It is not possible to say whether all end-devices of the network are using the same transmission rate. It is thus not sure whether all inter-message times become larger since the spreading of end-devices over time is unfortunately not performed such that the inter-message time is optimized. There could still be some cases in which two end-devices are located close to each other, although the chance is extremely small. Thus, when most of the end-devices use a transmission rate of once every 30 minutes, it is expected that no significant performance reduction will be present if the second downlink addressing cycle is not performed earlier.

Finally, it can be concluded that the proposed scheduling system can be used as even in the worst-case situation, a change of collisions is still relatively low compared to a network in which end-devices send at random moments in time.

Chapter 7

Conclusion and discussion

The first step towards a smart system in which measurement and system data can be accessed remotely and system commands can be given, consists of investigating wireless communication techniques. Wireless communication protocols SigFox, LoRaWAN and NB-IoT are intensively reviewed. Based on the long range, low power, and low cost requirement of this project, LoRaWAN is chosen as wireless communication technology.

Now the wireless communication protocol is chosen, off-the-shelf hardware components are chosen to implement LoRaWAN. The communication modem RN2483 and ANT-8WHIP3H-SMA antenna are chosen to implement the functionality of the end device in the network. The RN2483 was chosen based on the low cost, low power, and easy implementation, as the LoRaWAN protocol is already on-board. The ANT-8WHIP3H-SMA antenna is chosen based on a good gain, its VSWR properties, and its radiation pattern. However, the real performance of the antenna must be tested in practice. Also, properties of other important components in LoRaWAN are reviewed. The characteristics of a gateway, network server, and application server are discussed and it is concluded that by using LoRaWAN, warning system capabilities can be realised.

Furthermore, it is discussed how transmission parameters of a LoRaWAN end device can be configured. Also, LoRaWAN specifications are treated. It is concluded that the lowest power consuming device class A is used for this project. Also, uplink and downlink packet structures are explained and packet sizes of 11 and 10 bytes are chosen, respectively. With the use of these packet structures, the temperature measurement data of 5 connected temperature sensors and the energy level are sent with each uplink message. In a downlink packet, the time, an assigned time slot for the end device, the critical temperature, and whether the end device should perform measurements to protect crops or not. Finally, based on the specifications of a LoRaWAN class A device, the RN2483, and packet structure, a power breakdown per transmission is made. This power breakdown, consisting of multiple scenarios, resulted in a maximum energy consumption per transmission of 49.69 mJ.

Finally, a solution which implements scheduling in LoRaWAN is designed. The measurement system developed by the measurement and control group [6] requires a maximum transmission rate of once every 5 minutes. When 2000 LoRaWAN end devices would send randomly in these 5 minutes, it is found that a packet loss percentage of approximately 3% will be present. Thus, a scheduling system is designed to improve reliability. Due to time drifting of the clock of an end device, it is found that all end-devices need a frequent time synchronization message based on the transmission rate the end-device is using. It is found that by using the second receive window RX2 of a LoRaWAN class A end-device only, a network with scheduled end-devices can be realised with a significant reduction of collision chances.

Recommended future work

The first major point that has to be done in future work is actually implementing the system proposed in this thesis. Building such a system would cause new problems and challenges to be found which are not discussed in this thesis. In terms of the hardware, the antenna needs to be tested such that good estimations can be made about the performance. Also, a range test has to be done at a fruit orchard to get a correct estimation of the maximum distance. The RN2483 module must be installed and tested with the other hardware components to see if any problems arise from that. The last part would include implementing the scheduling algorithm in a network server and test the actual performance. Especially building the scheduling algorithm is seen as a interesting new opportunity to increase the reliability of LoRaWAN for large scale networks.

Furthermore, there are some other points of required future work that were not further investigated due to a lack of time. First of all, the current scheduling system is made such that it is optimised for a network with 2000 nodes that transmit every 5 minutes. A more general adaptive scheduling algorithm should be made that is optimised for end-devices that send less frequently than once every 5 minutes. A better distributing algorithm for spreading end-devices over time at lower transmission rates could increase network reliability even more since in that case, end-devices have fewer restrictions due to clock drift.
The simulation was done without clock drift. If this would have been implemented, a better estimation about the exact performance of the system could be made. However, to do this the exact behaviour of the clock drift should be found as only a worst case is now considered. Also, more system inaccuracies are unknown, such as the real processing time differences. Also, it was suggested that it might be possible to turn off the RN2483 in between transmissions. This would eliminate the power consumption of 5.28 μW in sleep mode. However, the feasibility should be investigated with respect to current and future firmware releases of the RN2483 and LoRaWAN specification updates.
Finally, it is important to note that possible effects of interference not belonging to the network were not investigated nor simulated. With increasing usage of IOT devices this can become a serious issue.

References

[1] Harold van der Meulen, “Fruitareaal en aantal bedrijven,” *https://www.agrimatie.nl/SectorResultaat.aspx ?subpubID=2232&sectorID=2237*, 2020. [Online]. Available: https://www.agrimatie.nl/SectorResultaat.aspx?subpubID=2232&sectorID=2237

[2] J. Rodrigo, “Spring frosts in deciduous fruit trees. Morphological damage and flower hardiness,” *Scientia Horticulturae*, vol. 85, no. 3, pp. 155–173, 2000.

[3] A. A. Ghaemi, M. R. Rafiee, and A. R. Sepaskhah, “Tree-Temperature Monitoring for Frost Protection of Orchards in Semi-Arid Regions Using Sprinkler Irrigation,” *Agricultural Sciences in China*, vol. 8, no. 1, pp. 98–107, 2009. [Online]. Available: http://dx.doi.org/10.1016/S1671-2927(09)60014-6

[4] A. C. Ribeiro, J. P. De Melo-Abreu, and R. L. Snyder, “Apple orchard frost protection with wind machine operation,” *Agricultural and Forest Meteorology*, vol. 141, no. 2-4, pp. 71–81, 2006.

[5] “Eurostat agricultural database.” [Online]. Available: https://ec.europa.eu/eurostat/web/agriculture/data/database

[6] M. Huiskes and M. Miao, “Autonomous Temperature Sensor for Smart Agriculture - Smart Measurement and Control,” 2020.

[7] H. A. Al-Kashoash and A. H. Kemp, “Comparison of 6LoWPAN and LPWAN for the Internet of Things,” *Australian Journal of Electrical and Electronics Engineering*, vol. 13, no. 4, pp. 268–274, 2016.

[8] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “A comparative study of LPWAN technologies for large-scale IoT deployment,” *ICT Express*, vol. 5, no. 1, pp. 1–7, 3 2019.

[9] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J. C. Prévotet, “Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility,” pp. 1561–1581, 2019. [Online]. Available: https://ieeexplore-ieee-org.tudelft.idm.oclc.org/abstract/document/8502812

[10] A. Villa-Henriksen, G. T. Edwards, L. A. Pesonen, O. Green, and C. A. G. Sørensen, “Internet of Things in arable farming: Implementation, applications, challenges and potential,” pp. 60–84, 3 2020.

[11] “Network coverage of Sigfox.” [Online]. Available: https://www.sigfox.com/en/coverage

[12] “GSMA | Deployment Map | Internet of Things.” [Online]. Available: https://www.gsma.com/iot/deployment-map/#deployments

[13] R. S. Sinha, Y. Wei, and S. H. Hwang, “A survey on LPWA technology: LoRa and NB-IoT,” *ICT Express*, vol. 3, no. 1, pp. 14–21, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.icte.2017.03.004

[14] B. Vejlgaard, M. Lauridsen, H. Nguyen, I. Z. Kovacs, P. Mogensen, and M. Sorensen, “Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT,” *IEEE Vehicular Technology Conference*, vol. 2017-June, pp. 8–12, 2017.

[15] ETSI, “ETSI Part 2 : Harmonised Standard covering the essential,” vol. 0, pp. 1–32, 2016.

[16] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad, and M. Ismail, “Energy-efficient wireless sensor networks for precision agriculture: A review,” *Sensors (Switzerland)*, vol. 17, no. 8, 2017.

[17] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, “Known and Unknown Facts of LoRa,” *ACM Transactions on Sensor Networks*, vol. 15, no. 2, pp. 1–35, 2019.

[18] “Narrowband-IoT: doorbreek de grenzen van IoT Ready?” Tech. Rep., 2017.

[19] J. M. Paredes-Parra, A. J. Garcia-Sanchez, A. Mateo-Aroca, and A. Molina-Garcia, “An alternative internet-of-things solution based on LOra for PV power plants: Data monitoring and management,” *Energies*, vol. 12, no. 5, 2019.

[20] Orange, “LoRa Device Developer Guide,” p. 42, 2016. [Online]. Available: https://partner.orange.com/wp-content/uploads/2016/04/LoRa-Device-Developer-Guide-Orange.pdf

[21] Semtech Corporation, “SX1301 LoRa chip datasheet,” no. June, pp. 1–40, 2017. [Online]. Available: www.semtech.com

[22] “ChirpStack Network Server - ChirpStack Network Server documentation.” [Online]. Available: https://www.chirpstack.io/network-server/overview/

[23] L. Alliance, “LoRaWAN™ 1.1Specification,” Tech. Rep., 2017.

[24] S. Kim, R. Vyas, J. Bito, K. Niotaki, A. Collado, A. Georgiadis, and M. M. Tentzeris, “Ambient RF energy-harvesting technologies for self-sustainable standalone wireless sensor platforms,” *Proceedings of the IEEE*, vol. 102, no. 11, pp. 1649–1666, 2014.

[25] Semtech, “An In-depth Look at LoRaWAN Class A Devices,” *Semtech Technique Paper*, no. November, pp. 1–12, 2019. [Online]. Available: https://lora-developers.semtech.com/uploads/documents/files/LoRaWAN_Class_C_Devices_In_Depth_Downloadable.pdf

[26] Microchip, *RN2483 LoRa™ Technology Module Command Reference User's Guide*, 2015. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf

[27] Semtech Corporation, “Datasheet SX1276/77/78/79 LoRa Transciever| Semtech,” no. January, p. 132, 2019. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001OKs/Bs97dmPXeatnbdoJNVMIDaKDIQz8q1N_gxDcgqi7g2o

[28] T. Bouguera, J. F. Diouris, J. J. Chaillout, R. Jaouadi, and G. Andrieux, “Energy consumption model for sensor nodes based on LoRa and LoRaWAN,” *Sensors (Switzerland)*, vol. 18, no. 7, pp. 1–23, 2018.

[29] Microchip Technology Inc., “RN2483 LoRa™ Datasheet,” pp. 1–22, 2017. [Online]. Available: http://ww1.microchip.com/downloads/en/devicedoc/50002346c.pdf

[30] Microchip Technology inc, “MCP9808 Maximum Accuracy Digital Temperature Sensor Datasheet,” pp. 1–54, 2011. [Online]. Available: ww1.microchip.com/downloads/en/DeviceDoc/25095A.pdf

[31] V. W. Heusinkveld, J. Antoon van Hooft, B. Schilperoort, P. Baas, M. c. t. Veldhuis, and B. J. van de Wiel, “Towards a physics-based understanding of fruit frost protection using wind machines,” *Agricultural and Forest Meteorology*, vol. 282-283, no. December 2019, p. 107868, 2020. [Online]. Available: https://doi.org/10.1016/j.agrformet.2019.107868

[32] G. Callebaut, “LoRaWAN Network Simulator.” [Online]. Available: https://github.com/GillesC/LoRaEnergySim

[33] Semtech, “How to Qualify a LoRaWAN ™ Device in Europe,” no. February, pp. 1–10, 2018.

Chapter A

General system design

Before the three subsystems are designed the implementation of the entire system is considered. The goal of the implementation is such that it fits the requirements of the entire system and that each subsystem can comply with their own requirements. When all these requirements are met, aspects that have to be reduced are manufacturing cost, deployment cost, and maintenance cost. Furthermore, durability has to be taken into consideration. To prevent further confusion, a single node is regarded as a subsystem with one wireless communication module, one microcontroller, and one energy harvesting module. The possible implementations of the system that were considered to fit into the requirements of the system are the following:

- **Tree multi node:** An implementation where every single node has one sensor. As a result, a tree contains 5 separate nodes on one tree. These multiple nodes are placed at different heights in the tree. This is a multi node implementation in a tree.
- **Tree single node:** A system where each node consists of a single MCU, wireless communication module, and energy harvester, while multiple sensors are connected to this single MCU. All sensors are placed in a tree at different heights.
- **Pole multi node:** A system where each node contains a single sensor. Multiple nodes are placed on different heights on a pole.
- **Pole single node:** A system where each node has multiple connected sensors and each node is integrated on a pole with the sensors at different heights.

To come to a design decision each option is weighted in the sections below. Based on this weight and the importance of each aspect an end decision is made which implementation is ultimately chosen.

Power consumption

The main power consumer is the wireless communication module. In a single node implementation, four wireless communication modules and MCUs are spared and thus this implementation is more preferable over a multi node implementation. The power consumption does not differ between a pole or tree implementation.

Cost

By implementing a single node structure, four wireless communication modules, four microcontrollers, and four energy harvesting modules are spared compared to a multi node structure. This gives a single node structure more advantages. The purchase of a pole gives a tree implementation preference compared to the pole implementation, although this can be offset by the lower installation cost.

Durability

Durability assesses the lifetime of the system. A pole implementation is considered more durable than a tree implementation since it has a more robust structure than the more fragile branches of a tree. For the pole implementation, it is difficult to assess whether a multi node or a single node implementation is better. However, in a single node implementation, fewer components are used and thus the chance of component failure is lower. On the other hand, in a single node implementation, temperature sensors are connected by wire and it depends on the type of integration whether this can have a significant negative impact on the device its lifetime.

Deployment

The deployment of the system assesses how easily the system can be installed. A significant advantage of a pole implementation is that it can be prefabricated and quickly installed by only planting the pole. For the tree, on-sight installation has to be performed which can be quite work-intensive, since sensors have to be installed on a certain height. Regarding a multi node or single node solution, the only main difference is the wiring which comes with a single node system. This is concerned to be more dramatic in a tree than on a pole since the wiring of prefabricated poles can be implemented more easily.

Placement consistency

The consistency of the system assesses how uniform the heights of the sensors are for each of the trees/poles. For an implementation with a pole the distance between sensors can easily be made very consistent, whereas for the implementation in a tree, heights can be slightly different due to the fact that trees have different shapes. For this factor, it does not matter whether a single-node or multi-node implementation is chosen.

Efficiency

The efficiency of the system describes how efficiently the system can harvest energy and transmit signals. For a pole implementation with the antenna on top, the transmitted signal likely comes across fewer obstacles. Also with the solar panel on top of a pole instead of on a tree, the panel will receive a lot more solar energy due to the fact that there is less shadow from the three itself. The difference between a multi-node implementation and single-node implementation also gives rise to different signal transmission- and energy harvesting efficiency. In the multi-node case, the solar panels are positioned on different heights. The nodes that are positioned higher will likely receive more solar radiation than lower positioned nodes. Furthermore, transmitted signals from the nodes that are positioned lower will likely come across more obstacles.

Flexibility

The flexibility of the system describes the ease of changing the number of sensors at a location. For a multi node pole implementation, the amount of sensors is changed with more ease than for a single node implementation. This is because, for a single node implementation, a sensor should be added or removed from the MCU, while for a multi node implementation only a complete module should be placed or removed without the need of connecting or disconnecting wires. The same holds for a tree implementation.

Maintainability

An important measure for designing a system is minimizing the cost of maintenance after the system has been designed to reduce further costs and time. To maintain the system where each node has a single sensor greatly increases the amount of hardware that can fail and needs to be maintained. The time it takes to replace a node can be neglected in the comparison between multi and single node as it will have much less influence on the total maintenance cost than the total amount of hardware failures. To compare the tree and pole solution it comes to the same comparison as with deployment namely disassembling the node and installing it again.

Expandability

By implementing a single-node structure, more nodes can be added before the communication system gets saturated. This because of the limitation in the number of wireless communication nodes that can be used for a certain gateway. By implementing a multi-node structure, the maximum amount of nodes gets reached more easily, and so the size of the sensor network becomes more limited.

Final decision

Table A.1: Decision matrix for a tree or pole and multi- or single node implementation

Criterion	Weight	Tree multi node		Tree single node		Pole multi node		Pole single node	
		Score	Weighted	Score	Weighted	Score	Weighted	Score	Weighted
Power consumption	5	3	15	10	50	3	15	10	50
Cost	7	6	42	10	70	5	35	9	63
Durability	9	4	36	3	27	9	81	8	72
Deployment	2	4	8	3	6	10	20	10	20
Placement consistency	2	6	12	6	12	10	20	10	20
Efficiency	5	1	5	5	25	3	15	10	50
Flexibility	3	8	24	2	6	8	24	2	6
Maintainablity	7	1	7	8	56	2	14	9	63
Expandability	2	2	4	10	20	2	4	10	20
			153		272		228		364

From Table A.1, it is obvious that a single node pole implementation is the best option for this application. The sub modules in this project are designed according to this specific implementation.

Chapter B

LoRaWAN

B.1 Frequency band regulations

In Table B.1, the European Standards for Short Range Devices listed by the ETSI Technical Committee (EN 300-220) are given. In Table B.2, the LoRaWAN frequency band regulations for deploying an end device in Europe are listed.

Table B.1: Table with ETSI regulations [15]

K	863 MHz to 865 MHz	25 mW e.r.p.	≤ 0,1 % duty cycle or polite spectrum access	The whole band except for audio & video applications limited to 300 kHz		46a
L	865 MHz to 868 MHz	25 mW e.r.p.	≤ 1 % duty cycle or polite spectrum access	The whole band		47
M	868,000 MHz to 868,600 MHz	25 mW e.r.p.	≤ 1 % duty cycle or polite spectrum access	The whole band		48
N	868,700 MHz to 869,200 MHz	25 mW e.r.p.	≤ 0,1 % duty cycle or polite spectrum access	The whole sub-band		50
P	869,400 MHz to 869,650 MHz	500 mW e.r.p.	≤ 10 % duty cycle or polite spectrum access	The whole band		54
P	869,700 MHz to 870,000 MHz	5 mW e.r.p.	No requirement	The whole band	Audio and video applications are excluded.	56a
Q	869,700 MHz to 870,000 MHz	25 mW e.r.p.	≤ 1 % duty cycle or polite spectrum access	The whole band	Analogue audio applications are excluded. Analogue video applications are excluded.	56b

Table B.2: Table with LoRaWAN regulations [33]

Band per EC Decision [5]	Frequency Range (MHz)	Duty Cycle Limit[1]	Maximum e.r.p.	Implementation Deadline	Comment in regards to LoRaWAN™
48	868 – 868.6	1 %	25 mW	July 1st 2014	Join channels, mandatory
50	868.7 – 869.2	0.1 %	25 mW	July 1st 2014	Low Duty Cycle
54	869.4 – 869.65	10 %	500 mW	July 1st 2014	May not be used for uplink, only downlink
56b	869.7 - 870	1 %	25 mW	July 1st 2014	
46a	863 – 865	0.1 %	25 mW	January 1st 2018	New band, low duty-cycle
47	865 – 868	1 %	25 mW	July 1st 2014	
47b	865 – 868	2.5 %	500 mW	January 1st 2018	Only specific channels[2]

[1] Averaged over 1 hour and summing the time on air of all channels declared within the sub-band

[2] Transmissions only permitted within the bands 865,6- 865,8 MHz, 866,2-866,4 MHz, 866,8-867,0 MHz and 867,4- 867,6 MHz. Adaptive Power Control (APC) required.

B.2 Data rates in Europe

In Table B.3 below, the Data Rates (DRs) DR0 to DR5 which can be used in Europe are listed. All these data rates have a Bandwidth (BW) of 125 kHz. The maximum application payload for each DR is listed.

Table B.3: LoRaWAN data rates in Europe

Data rate (DR)	Modulation	SF	BW [kHz]	bps	max. payload
0	LoRa	12	125	250	51
1	LoRa	11	125	440	51
2	LoRa	10	125	980	51
3	LoRa	9	125	1760	115
4	LoRa	8	125	3125	222
5	LoRa	7	125	5470	222
6	LoRa	7	250	11000	222
7	FSK	-	150	50000	222

B.3 LoRa packet time on air

The Time On Air (TOA) of a LoRa packet can be calculated following the procedure described below [27]. The symbol time is given by:

$$T_{sym} = \frac{2^{SF}}{BW}$$

The total time for the preamble is given by:

$$T_{preamble} = (n_{preamble} + 4.25)T_{sym}$$

The payload size is given by:

$$N_{payload} = 8 + max(ceil(\frac{8PL - 4SF + 28 + 16 - 20H}{4(SF - 2DE)})(CR + 4), 0) \quad \text{(B.1)}$$

The symbols represent the following:

- PL: The number of payload bytes.
- SF: The spreading factor.
- H = 0 when the header is enabled and H = 1 when no header is present.
- DE = 1 when the low data rate optimization is enabled , DE = 0 for disabled.
- CR: The coding rate from 1 to 4.

For the calculations in this thesis, the header is always enabled thus H=0. Also, CR=1 which represents a coding rate of $\frac{4}{5}$ is used. Low data rate optimization is only used for SF11 and SF12, as required by the LoRaWAN specification [23].

The total Time On Air (TOA) is then given by:

$$TOA = T_{preamble} + N_{payload} * T_{sym}$$

B.4 Time slots

In Figure B.1, it is shown that when the end-devices are distributed equally among all SFs, the inter-message free channel time drops with higher SFs. This is because higher SFs have higher TOA of an 11 byte message.

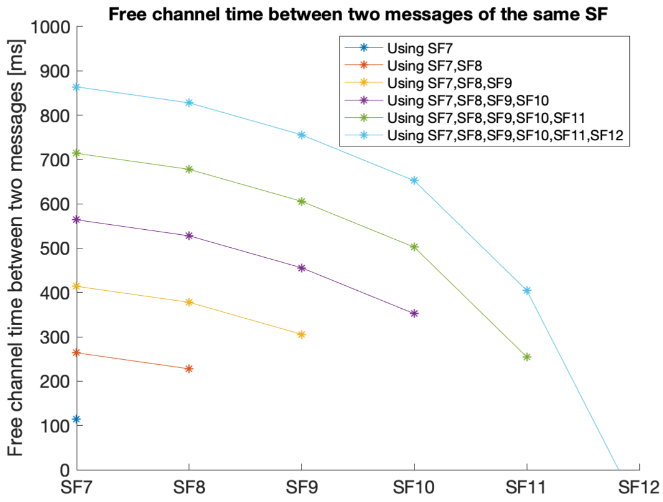


Figure B.1: Free channel time when equally distributing end devices

However, it could be that a certain pattern of time slots is not advantageous for RX1 downlink messaging since RX1 is always 1 second after a transmission. In Figure B.2 below, the result of using RX1 with the inter-message time shown in Figure B.1 is given. In that case, all nodes are spread among all SFs equally. It can be seen that using this method of scheduling, collisions with RX1 downlink messages and uplink messages might occur for end devices using SF9 and SF10, when the other uplink sending device is on coincidence in the same frequency channel. Also, SF8 devices have a chance of shifting into an uplink time slot due to time drift the MCU clock.

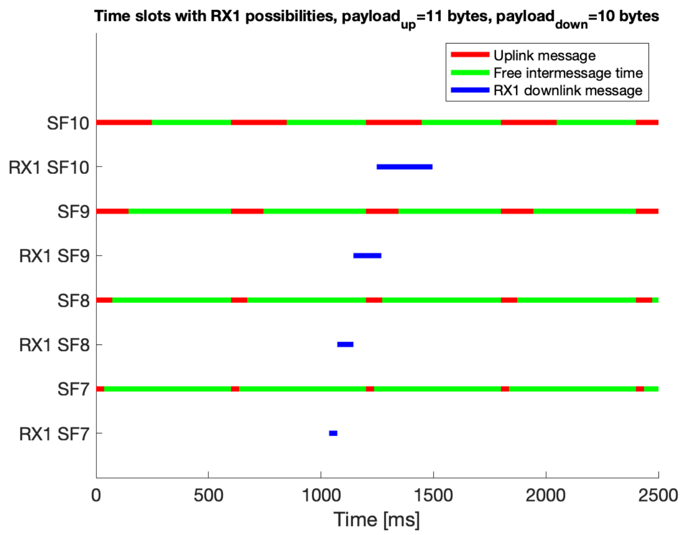


Figure B.2: Time slots and RX1 scheduling with nodes equally distributed over SFs