UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

University of Padova

Department of Information Engineering

Bachelor thesis in
Computer Engineering

An LPWAN MAC protocol for agricutural applications

Supervisor:
Prof. Stefano Tomasin

Candidate:
Giulio Codutti
2008795

Year 2022/2023

Abstract

Low Power Wide Area Networks (LPWAN) are getting very popular these days in Internet of Things (IoT) applications thanks to their capability of both consuming low amounts of power and of covering long distances. This technology is widely used in the 4^{th} industrial era for manufacturing, health care, and automation in general.

This thesis has the objective to propose a Media Access Control (MAC) protocol called Bacco. It is based on LoRa modulation and has a narrow focus on agricultural applications, where achieving high power efficiency is crucial due to the lack of reliable power sources. Another aspect taken into consideration is the cost effectiveness of the devices required to develop a functional network.

First, the thesis establishes an introduction of LoRa and LoRaWAN; then the requirements for a MAC protocol used in LPWANs will be discussed. After that, there will be a description of the Bacco protocol itself, alongside with some example applications of it.

Sommario

Le reti Low Power Area Network (LPWAN) stanno prendendo piede oggigiorno nel mondo dell'Internet of Things (IoT) grazie al loro basso consumo energetico e alle ampie distanze che possono coprire. Questa tecnologia è un caposaldo dell'industria di quarta generazione, soprattutto negli ambiti di manifattura, assistenza sanitaria e in generale dell'automazione.

Questa tesi ha l'obiettivo di proporre un protocollo Media Access Control (MAC), chiamato Bacco. Questo sfrutta la modulazione LoRa e si rivolge a applicazioni in ambito agricolo, dove è cruciale raggiungere un'alta efficienza energetica data la mancanza di fonti energetiche affidabili. Un altro aspetto che viene considerato è il costo dei dispositivi rischiesti per sviluppare una rete funzionale.

Inizialmente la tesi si occuperà di dare una breve introduzione a LoRa e LoRaWAN, per poi discutere i requisiti di un protocollo MAC per LPWAN. Successivamente, verrà data una descrizione del funzionamento di Bacco, accompagnata da alcuni esempi applicativi.

Glossary

GSM Acronym for “Global System for Mobile Communications”, it’s a 2nd generation mobile communication standard, see [1] for more information.

LTE Acronym for “Long Term Evolution”, it’s a 4th generation mobile communication standard, see [2] for more information.

FTP Acronym for “File Transfer Protocol”, built on top of TCP, see [3] for more information.

VHF Acronym for “Very High Frequency”, it refers to the radio frequency band between 30 and 300 MHz.

PHY Stands for physical layer protcol.

Contents

Chapter 1

Introduction

Place introduction here

Chapter 2

Bacco protocol

The goal of this chapter is to give a description of the *Bacco* protocol and to discuss the implementation choices that were made in order to deploy it. This is achieved using a top-down ordering for the level of detail, meaning that the overview is presented before the specifics.

2.1 Overview and case of study

We will now start describing a simple scenario that makes use of the protocol to better understand the integration of Bacco into an LPWAN. The network is built upon 4 categories of devices:

- Sender node - collects data and sends it to the gateway

- Repeater node - listens to the incoming messages from Senders and forwards them to a Gateway [1]

- Gateway node - collects data coming from the sender nodes and sends it to the web server. This node has the role of coorinating and syncronizing Sender nodes. In the example shown in figure 2.1, this is achieved using the FTP protocol over a GSM or LTE mobile network. It can be optionally configured to perform pre-processing operations (e.g. filtering, smoothing, interpolation ...) on the incoming data

[1] The use of Repeaters where physical obstacles compromise the integrity of the signals is of very high relevance in agricultural contexts, since natural barriers such as hills can easily block VHF radio signals.

- WEB SERVER - receives data coming from the Gateways, elaborates it and makes it available through a web application. Note that the scheme of communication involving this device is not covered by Bacco

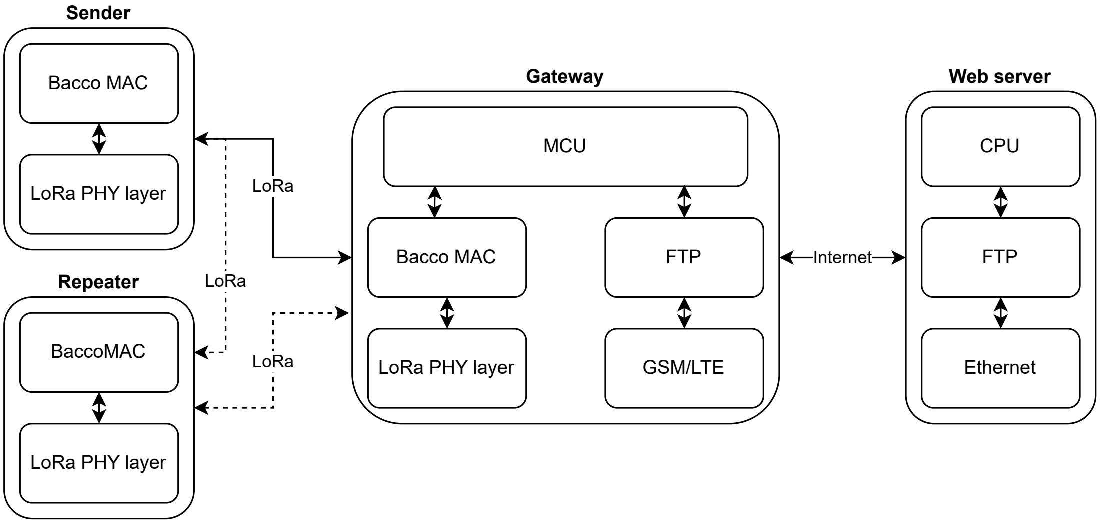

Figure 2.1: Schematic representation of an example network using Bacco

2.2 Topology

The network has a star-of-stars topology, in which the zeroth level is occupied by the Web server, the first level by the Gateways and Repeaters, and the second level contains the Senders. Figure 2.2 shows the type of devices that are involved and their communication schema.
The structure is equivalent to a tree, hence we can define a hierarchy for the nodes. The root node is the central web server and its children nodes are the Gateways. All sender nodes are children of either a Gateway or a Repeater and have no chlidren, so they correspond to the leaves of the tree.

2.3 Addressing

It is crucial to identify each Sender node in order to contextualize the messages coming to the Gateway node. This is achieved by assigning them a unique identifier, represented by a natural number in the range $[1, 254]$. Address 0 is reserved for the Receiver and address 255 is used as a globally invalid address. This fact

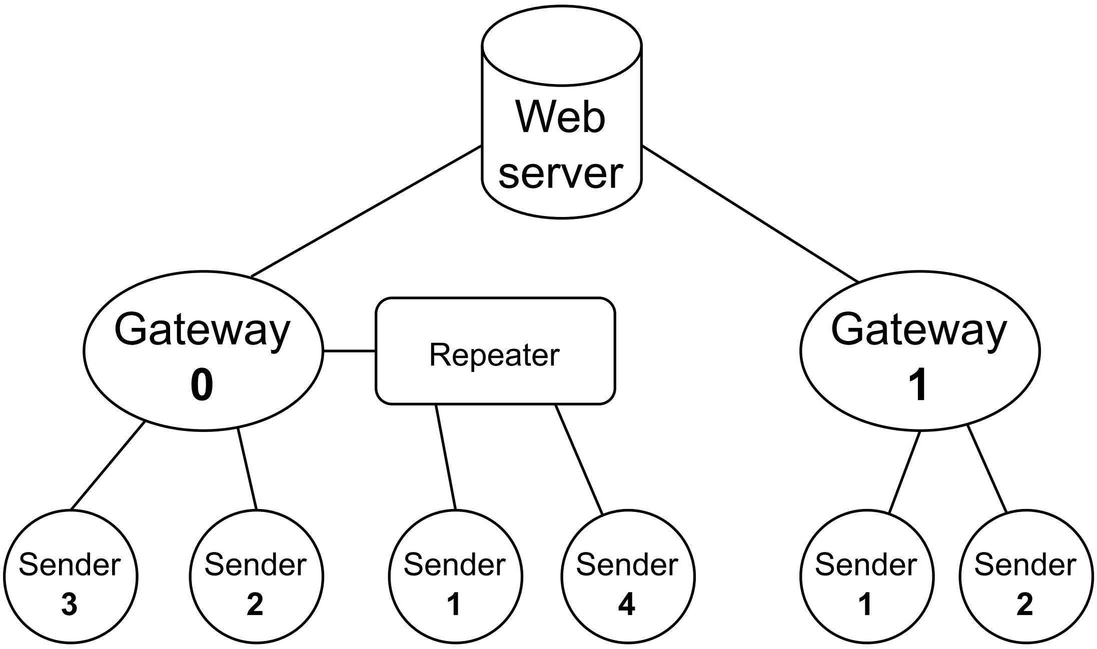

Figure 2.2: Example network topology

limits the number of Sender nodes connected to a single Gateway to 254 [2]. If necessary, the network can scale up by using additional Gateways. Note that since Repeaters do not produce messages themselves nor they modify the forwarded ones, they will not be given an identifier.

2.4 Interference mitigation

The LoRa PHY protocol specification does not fully cover the matter of sharing the communication link between multiple devices, thus it is necessary to define methods for doing so, in order to minimize interference and achieve a realiable exchange of information. Different techniques are applied in the domain of both time and frequency.

2.4.1 Channel activity detection

Channel Activity Detection, or CAD, is a feature available with most LoRa transcievers[5]. In this mode, the LoRa node listens for any transmission on a specific

[2] This choice is influenced by the current Italian regulations TODO: Scrivi e inserisci citazione a paragrafo su regolamentazione e calcolo numero massumo di devices **CITAZIONE PARAGRAFO REGOLAMENTAZIONE E CALCOLO MASSIMO NUMERO DI** [4] on duty cycle for the 868MHz band and the fact that most agricultural contexts do not require a huge amount of sensors

frequency and, if it detects a signal, an interrupt is returned to the MCU. This presents a possible Carrier Sense Multiple Access mechanism, also called CSMA. Bacco does not make use of CAD for data packets, but it enables it in specific situations such as network discovery (discussed in section 2.5).

2.4.2 IQ inversion

IQ inversion is a LoRa primitive that makes it possible to have 2 types of transmissions on the same frequency and spreading factor, that are easily distinguishable for a receiver. The name stands for *In-phase / Quadrature*, and it usually refers to signals that are out of phase from each other by $\frac{\pi}{4}$ rad. Despite that, LoRa uses the term to describe 2 signal with inverted chirp direction, namely up-chirp and down-chirp (see figure 2.3).

Bacco uses this technique to discriminate between uplink messages (i.e. from Sender to Gateway/Repeater and fom Repeater to Gateway) and downlink messages (i.e. from Gateway to Sender/Repeater and from Repeater to Sender). This implies that Sender nodes and Gateway nodes are not able to communicate with other devices of the same category (e.g. a Sender would not detect any transmission coming from a Sender).

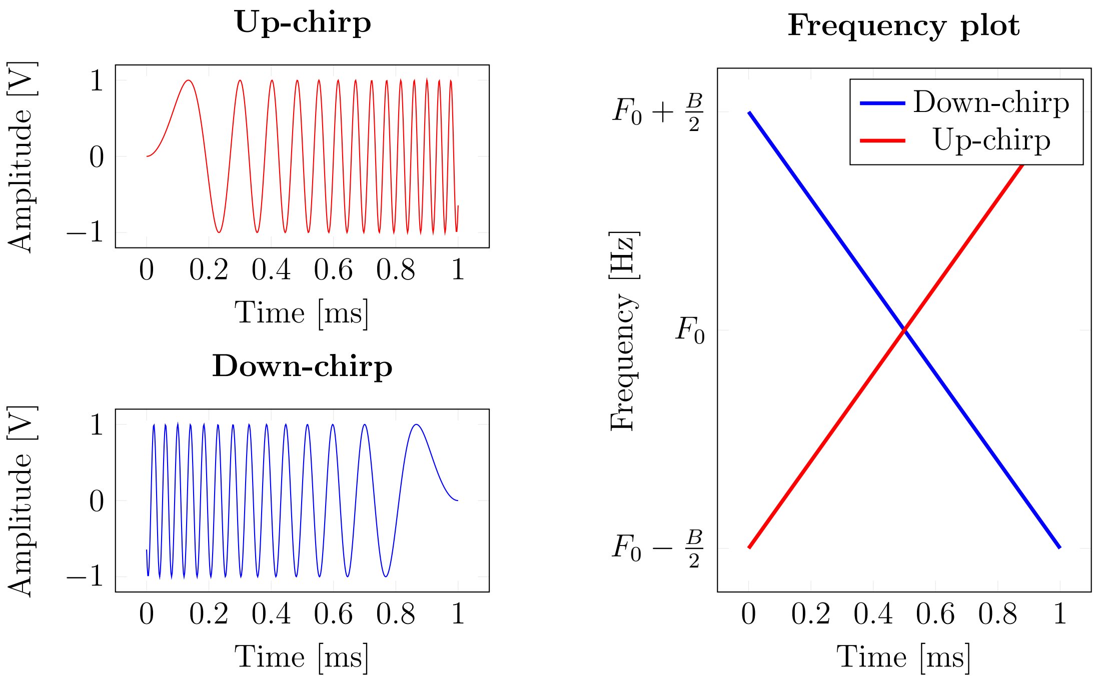

Figure 2.3: IQ inversion

2.4.3 Subnetting

The LoRa procol supports a wide range of carrier frequencies in the VHF spectrum [3].

Bacco exploits this fact to build sub-networks that operate at different frequencies, and thus it achieves a communication with very low interference between different the independent parts. All the Sender nodes connected to a Repeater and the Repeater itself operate at a frequency that's different from the frequency used by the Gateway and other Repeaters. Figure 2.4 shows a schematic representation. This technique is particularly useful to solve the problem of bouncing, that occours when a message is sent back to the original Repeater from another Repeater.

It is necessary to define a set of frequencies at which the nodes operate. This choice is done based on current regulations in the specific country of operation; this thesis will focus on Italian and/or European standards. 868 MHz is the base frequency at which the Gateway node operates. The set of 11 frequencies of operation is so composed:

$$f = \{f_k : f_k = 868\text{MHz} + k \times 125\text{KHz}, k \in \{0..10\}\} \tag{2.1}$$

This implies that a Bacco network can have up to 10 Repeaters operating at different frequencies, however if more coverage is needed, it is possible to have multiple Repeater nodes working at the same frequency, given that they are not in reach with each other.

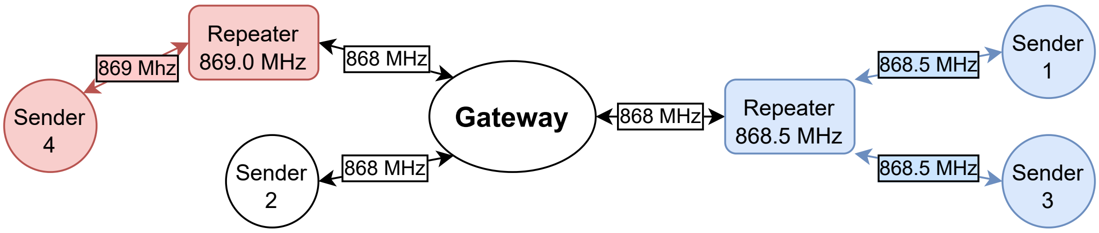

Figure 2.4: Example network with repeaters in subnets

2.4.4 Distribution of transmission activity

The Bacco protocol distributes radio activity over time with defined frames reserved for each Sender, using an approach that was first introduced by the Alo-

[3] For reference, the SX1262[6] transciever features a continuous frequency coverage from 150 MHz to 960 MHz

haNet[7] protocol. The frames are equally distributed between all the Senders and the start of each of them is function of the identifier. The time delay between 2 consecutive transmissions from a same Sender is a constant value and it is called cycle. At the end of a cycle, some time is reserved for the Gateway to upload the collected data. Figure 2.5 shows a schematic representation of the time management used by Bacco. The cycle time is a user-defined parameter, and all the other values are calculated based on it as shown in table 2.1.

Parameter	**Value**
Gateway frame time	$0.2 \cdot C$
Sender frame time	$\frac{0.6 \cdot C}{N_{max}}$
Silence frame time	$\frac{0.2 \cdot C}{N_{max}}$

Table 2.1: Time parameters calculation. C is the cycle time and $N_{max} = 254$ is the maximum number of Senders

Compensating clock drift

All the devices in the network need to transmit during their assigned frame for the protocol to be effective; this means that all the clocks of the devices are required to be in sync. This is hard to achieve without dynamic recalibration, because commercial oscillators cannot provide a constant frequency source due to manifacturing imprecisions, temperature gradient etc... .
In order to deal with with this problem, Bacco assigns the Gateway node the role of coordinating the network timings through the dispatch of downlink messages containing the network timestamp. A downlink message is sent as soon as the Gateway receives an uplink message that exceedes its correct time frame. Figure 2.5 shows this behavior during the transmission involving Sender2. Additionally, the Gateway sends at least 1 downlink message every 10 uplink messages from a same Sender, to signal that the connection is still active.

2.5 Network discovery

When a Sender node is first started, it needs to decide at what frequency it has to operate for minimizing the power needed to reach a Repeater or Gateway. In order to do this, Bacco introduces method 1 for scanning nearby devices and selecting

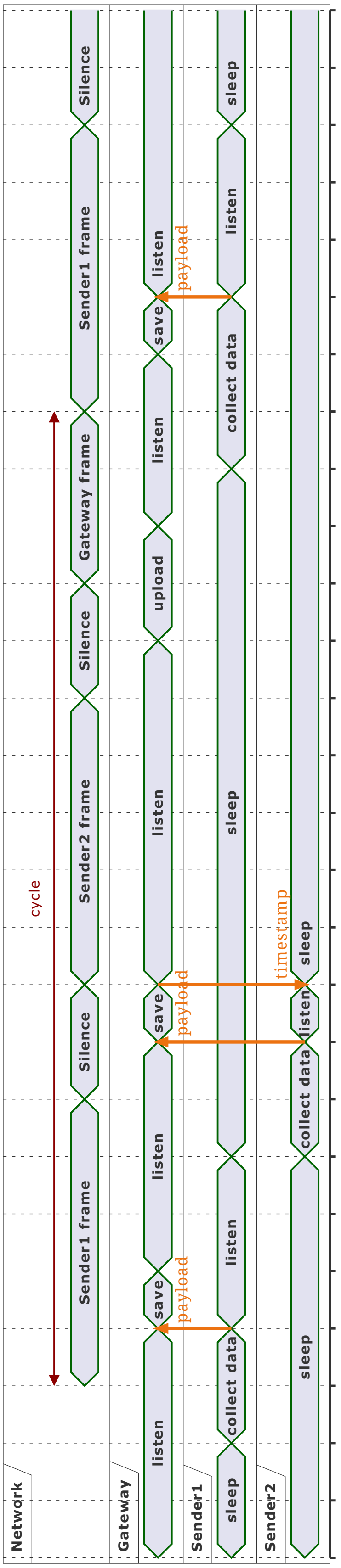

Figure 2.5: Timing diagram

the most suitable. It tries to establish a communication with Repeaters and Gateways throughout all the available frequencies by sending a particular type of message that triggers a SYN/ACK response. CAD will be used by the Sender to not interfere with ongoing communications; this is because the board does not yet have an allocated time frame and thus cannot decide when to transmit otherwise.

Algorithm 1 Network discovery

```
rssi_values ← $[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$
while all rssi_values are equal to 0 do
    for $k$ from 0 to 10 do
        $f_k \leftarrow 868 \times 10^6 + k \times 125 * 10^3$
        for $i$ from 0 to 10 do
            do
                sleep for 1 second
                enter CAD mode at frequency $f_k$ for 1.5 seconds
            while activity detected by CAD
            send sensing message
            enter receive mode for 3 seconds
            if received SYN/ACK then
                rssi_$values\,[k]$ ← current rssi
                break
            end if
        end for
    end for
end while
return $868 \times 10^6 +$ argmin(rssi_values) $\times 125 * 10^3$
```

2.6 Network joining

When a Sender node needs to connect to a Gateway node for the first time, it does not yet have an identifier nor is in sync with the network. The procedure to achieve that will be called the joining process. Note that we assume that the Sender node has already selected its frequency of operation, as descibed in section 2.5.

We will ignore the act of forwarding made by any optional Repeater node, as in this case it does not affect the content of the messages, but note that delay and error rate would raise in that situation. All the messages sent from Sender and Receiver make use of CAD to make sure the channel is free before the actual transmission; this step will be omitted in the description for brevity. First, the

Sender transmits a SYN message to the Gateway and waits for a SYN/ACK response for 3 seconds. If no message is received, another SYN message is sent for a maximum of 10 times. After that, the Gateway waits for 3 seconds for an ACK message from the Receiver, and if no message is received it will try again for a maximum of 10 times. The SYN/ACK message contains the timestamp of the network as well as the assigned identifier. If the maximum number of iterations is reached in any of the steps, the process starts again after 30 minutes. Figure 2.6 show a schematic representation of the process.

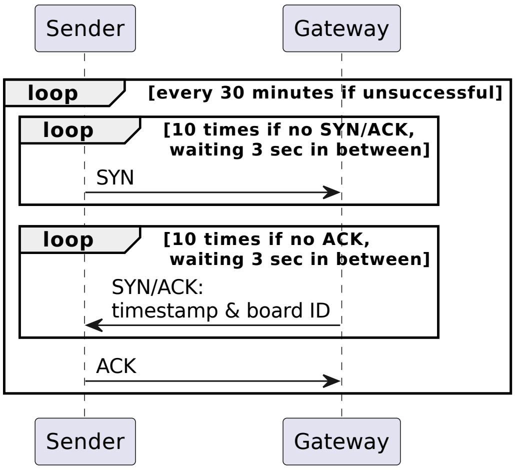
Sender
Gateway
loop
[every 30 minutes if unsuccessful]
loop
[10 times if no SYN/ACK, waiting 3 sec in between]
SYN
loop
[10 times if no ACK, waiting 3 sec in between]
SYN/ACK: timestamp & board ID
ACK
Sender
Gateway

Figure 2.6: Network joining process

2.7 Downlink commands

In some situations (e.g. 2.8), it is required to be able to change the behavior of the network dynamically and in a reliable way. Bacco achieves this by exhanging specialized kind of packets that contain a command. Each command message needs to be acknowledged by the Sender to the Gateway/Repeater in a similar way as done during the network joining procedure. Figure 2.7 offers a graphical representation of the process.

The following commands are defined by the protocol, and each of them is associated to an opcode as shown in table 2.2:

- SHUTDOWN - If this command is sent and processed successfully, the Sender goes to sleep indefinetly until a manual reset is invoked by pressing a physical button
- ENTER QUIESCENT MODE - If this command is processed successfully, the Sender

stops to send data, but it keeps listening for incoming messages/commands during its time frame

- WAKEUP - If this command is processed successfully, the Sender enters normal/active mode and thus starts to transmit data
- INCREASE TRANSMISSION POWER - If this command is processed successfully, the Sender increases its transmit power by 3dBm
- DECREASE TRANSMISSION POWER - If this command is processed successfully, the Sender deceases its transmit power by 3dBm

Command	**Opcode as 7 bit unsigned integer**
Shutdown	0
Enter quiscent mode	1
Wakeup	2
Increase transmission power	3
Decrease transmission power	4
Reserved for later use	[4-50]
User defined	[51-127]

Table 2.2: Table of opcodes

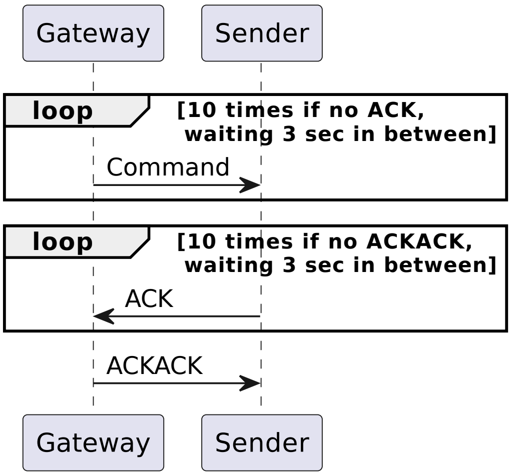
Gateway
Sender
loop
[10 times if no ACK, waiting 3 sec in between]
Command
loop
[10 times if no ACKACK, waiting 3 sec in between]
ACK
ACKACK
Gateway
Sender

Figure 2.7: Command sending process

2.8 Transmission power adaption

Since Senders can be placed at different distances from a Gateway or Repeater, it is useful to optimize the power used by the node to transmit. The default value for the transmission power is equal to P_0; starting from that, the network will automatically drift towards a more suitable value according to the following triggers:

- If a Sender has not received any downlink messages during the last 20 frames, then its transmission power will be increased by P_{step}
- If 9 out of the last 10 messages received by a Repeater or a Gateway satisfy RSSI $>$ RSSI$_{high}$ and SNR $>$ SNR$_{high}$, then a command is sent to the Sender telling to decrease its transmission power by P_{step}
- If 9 out of the last 10 messages received by a Repeater or a Gateway satisfy RSSI $<$ RSSI$_{low}$ or have not been received, then a command is sent to the Sender telling to increase its TX power by P_{step}.

The values for the used parameters are shown here:

$$\begin{aligned} &P_0 = P_{max} = 14 \text{ dBm} \\ &P_{step} = 3 \text{ dBm} \\ &\text{RSSI}_{low} = -115 \text{ dBm} \\ &\text{RSSI}_{high} = -60 \text{ dBm} \\ &\text{SNR}_{low} = -7 \text{ dBm} \end{aligned} \tag{2.2}$$

Gateways and Repeaters always operate at P_0, since power efficiency is less critical than reliability for such nodes.

2.9 Packet format

In this section the bit format of the messages is shown. The analysis will be split betqween uplink packets and downlink packets.

Uplink packet format

An uplink message is sent from a Sender to a Gateway/Repeater or from a Repeater to a Gateway. It has a variable length and it is transmitted using up-chirps, i.e with IQ inversion disabled. Figure 2.8 shows its format.

0	8	16
Sender ID	Payload size	Payload

Figure 2.8: Uplink packet format

Downlink packet format

A downlink message is sent from a Gateway to a Sender/Repeater or from a Repeater to a Sender. It has a fixed length of 5 bytes and it is transmitted using down-chirps, i.e with IQ inversion enabled. Figure 2.9, 2.10, 2.11 show the general format, the timestamp format and the command format respectively.

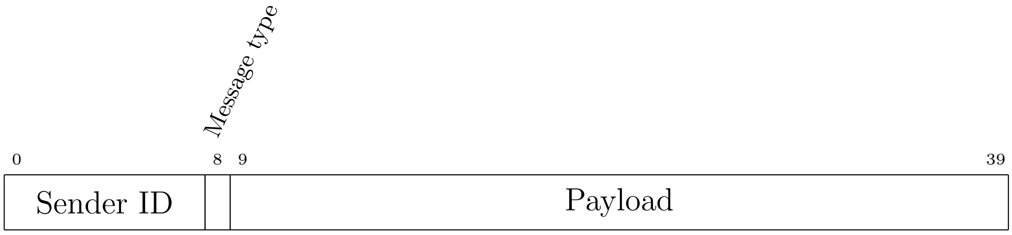

Figure 2.9: Downlink packet format

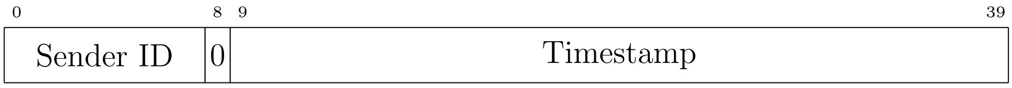

Figure 2.10: Downlink packet format of timestamps

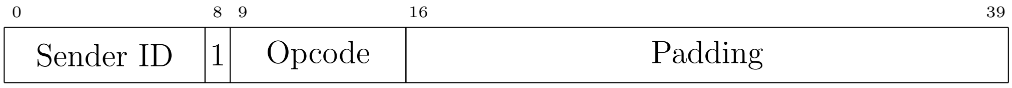

Figure 2.11: Downlink packet format for commands

Chapter 3

Performance

The goal of this chapter is ...

3.1 Time on air

3.1.1 Regulations

3.1.2 Lab tests

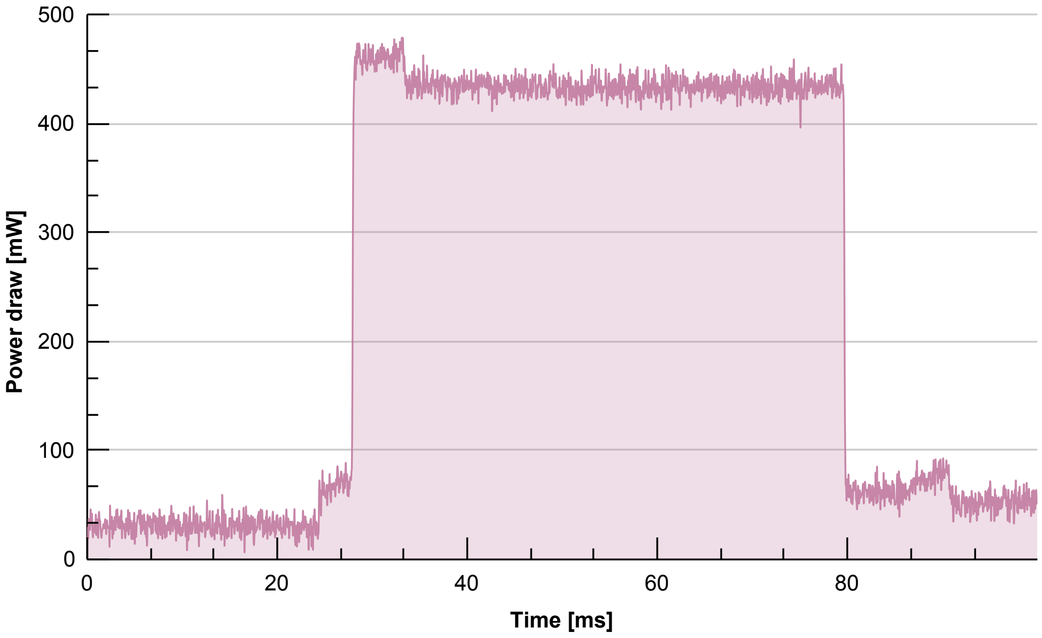

Figure 3.1: Power draw during transmission of a Bacco packet with payload size of 15 bytes, using SF7, 14dBm, 125kHz bandwidth

delta time is 51.6ms and total energy is 21.3mJ

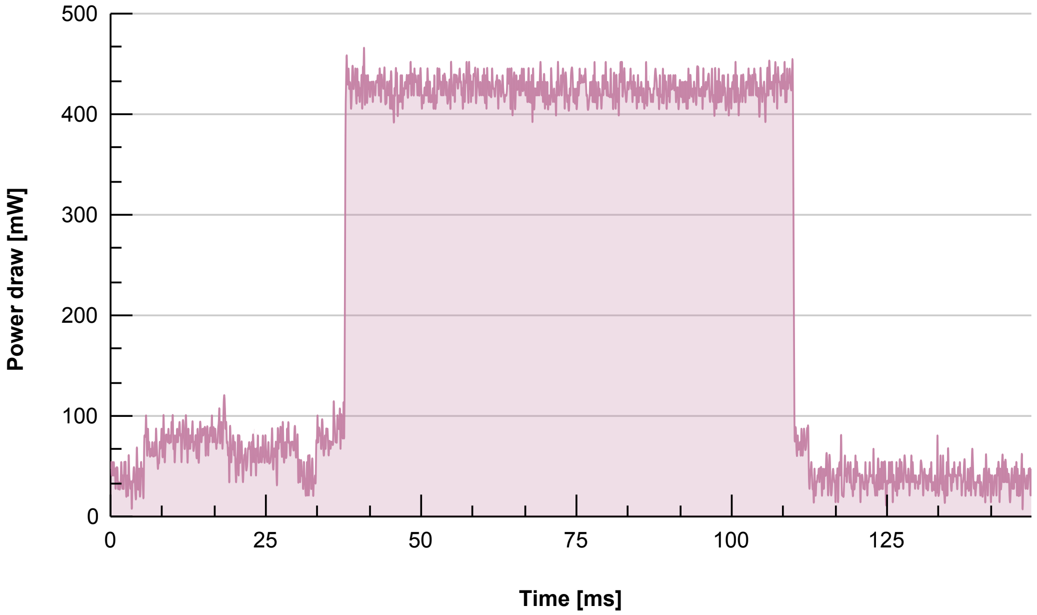

Figure 3.2: Power draw during transmission of a LoRaWAN packet with payload size of 15 bytes, using SF7, 14dBm, 125kHz bandwidth

delta time is 71.8ms and total energy is 30.8mJ

Bibliography

[1] V. K. Garg and J. E. Wilkes, *Principles and Applications of GSM.* Prentice Hall PTR, 1998, ISBN: 0139491244. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/552293.

[2] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "Lte-advanced: Next-generation wireless broadband technology [invited paper]", *IEEE Wireless Communications*, vol. 17, no. 3, 2010. DOI: 10.1109/MWC.2010.5490974.

[3] J. Postel and J. Reynolds, "File transfer protocol (ftp)", Network Working Group, RFC 959, 1985. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc959.

[4] *Elenco delle interfacce per gli apparati radioelettrici - apparati a corto raggio utilizzati per applicazioni generiche (es. telemetria, telecomandi, allarmi e simili)*, Feb. 2005. [Online]. Available: https://www.gazzettaufficiale.it/atto/serie_generale/caricaArticolo?art.versione=1&art.idGruppo=0&art.flagTipoArticolo=1&art.codiceRedazionale=05A00705&art.idArticolo=1&art.idSottoArticolo=1&art.idSottoArticolo1=10&art.dataPubblicazioneGazzetta=2005-02-12&art.progressivo=0.

[5] *Channel activity detection: Ensuring your lora packets are sent*, Semtech. [Online]. Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/channel-activity-detection-ensuring-your-lora-packets-are-sent/how-to-ensure-your-lora-packets-are-sent-properly/.

[6] *Sx1261/2 datasheet*, Rev. 2.1, Semtech, Dec. 2021. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/E0000000JelG/a/2R000000Un7F/yT.fKdAr9ZAo3cJLc4F2cBdUsMftpT2vsOICP7NmvMo.

[7] N. Abramson, "Development of the alohanet", *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 119–123, 1985. DOI: 10.1109/TIT.1985.1057021.