



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 agricultural 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<sup>th</sup> 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 oggi giorno 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 richiesti 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** Global System for Mobile Communications, 2nd generation mobile communication standard, see [1] for more information.

**LTE** Long Term Evolution, 4th generation mobile communication standard, see [2] for more information.

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

**VHF** Very High Frequency, it refers to the radio frequency band between 30 and 300 MHz.

**PHY** physical layer protocol.

**LPWAN** low power wide area network.

**IoT** Internet of Things, it refers to the concept of a worldwide network of interconnected sensors.

**LoRaWAN** LoRa Wide Area Network, it refers to an open network protocol built on top of LoRa.

**CAD** channel activity detection.

**CSMA** carrier sense multiple access.

**SYN** sync message.

**SYN/ACK** sync acknowledge message.

**ACK** acknowledge message.

**ACKACK** double acknowledge message.

**MCU** micro controller unit.

**IQ** in phase/quadrature.

# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Scenario And LoRaWAN Fundamentals</b>	<b>3</b>
<b>2 Bacco Protocol</b>	<b>5</b>
2.1 Overview And Case Of Study . . . . .	5
2.2 Topology . . . . .	6
2.3 Addressing . . . . .	6
2.4 Interference Mitigation . . . . .	7
2.4.1 Channel Activity Detection . . . . .	7
2.4.2 IQ Inversion . . . . .	8
2.4.3 Subnetting . . . . .	8
2.4.4 Distribution Of Transmission Activity . . . . .	9
2.5 Network Discovery . . . . .	10
2.6 Network Joining . . . . .	12
2.7 Downlink Commands . . . . .	12
2.8 Transmission Power Adaption . . . . .	13
2.9 Packet Format . . . . .	15
<b>3 Performance</b>	<b>19</b>
3.1 Time on air . . . . .	19
3.1.1 Regulations . . . . .	19
3.1.2 Lab tests . . . . .	19





# Introduction

Place introduction here



# **Chapter 1**

## **Scenario And LoRaWAN Fundamentals**

Insert cap 1 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 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

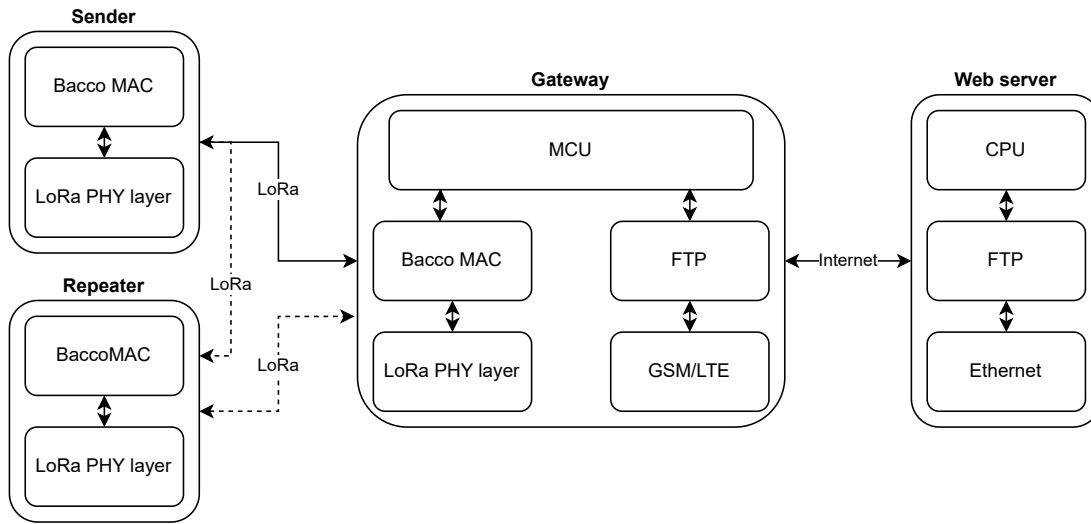
We will now start describing a simple scenario that makes use of the protocol to better understand the integration of Bacco into a low power wide area network (LPWAN). The network is built upon 4 categories of devices:

- **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; <sup>1</sup>
- **GATEWAY NODE** - collects data coming from the sender nodes and sends it to the web server. This node has the role of coordinating and synchronizing Sender nodes. In the example shown in Figure 2.1, this is achieved using file transfer protocol (FTP) over a global system for mobile communications (GSM) or long term evolution (LTE) mobile network. It can be optionally configured to perform pre-processing operations (e.g. filtering, smoothing, interpolation ...) on the incoming data;

---

<sup>1</sup>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 very high frequency (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 Gateways and Repeaters, and the second level contains the Senders. Figure 2.2 shows the types of devices that are involved and their communication scheme.

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 children, so they correspond to the leaves of the tree.

## 2.3 Addressing

It is crucial to identify each Sender node 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 limits the

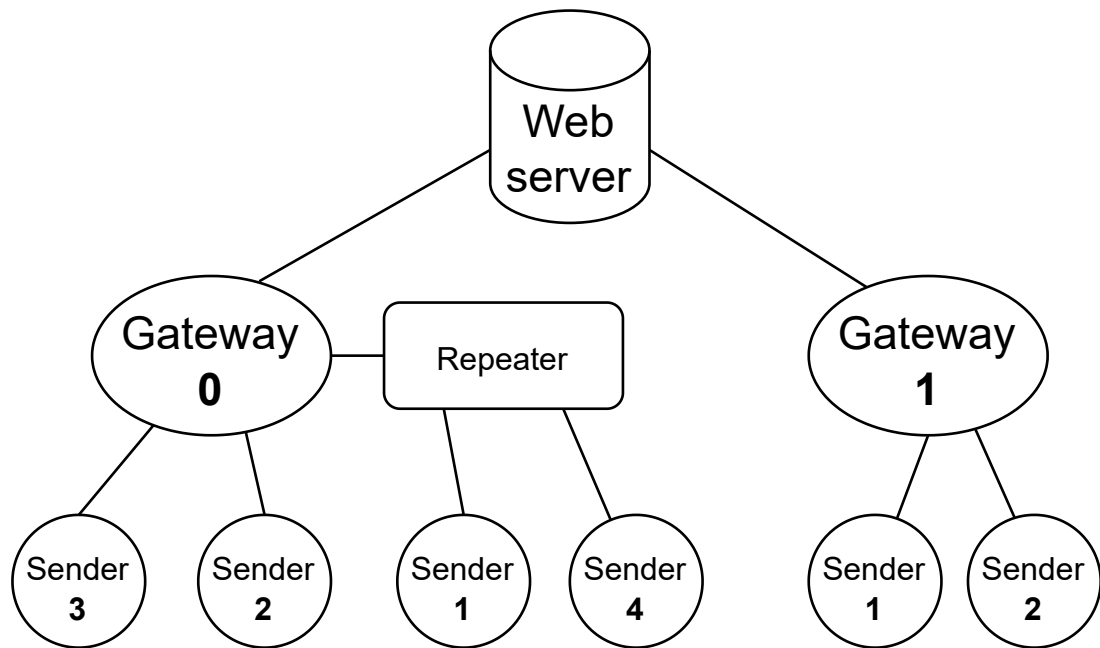


Figure 2.2: Example network topology

number of Sender nodes connected to a single Gateway to 254<sup>2</sup>. 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, to minimize interference and achieve a reliable exchange of information. Different techniques are applied in the domain of both time and frequency.

### 2.4.1 Channel Activity Detection

Channel activity detection (CAD) is a feature available for most LoRa transceivers [5]. In this mode, the LoRa node listens for any transmission on a specific

<sup>2</sup>This choice is influenced by the current Italian regulations  
 TODO: Scrivi e inserisci citazione a paragrafo su regolamentazione e calcolo numero massimo 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 micro controller unit (MCU). This presents a possible carrier sense multiple access (CSMA) mechanism.

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

In phase/quadrature (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 IQ usually refers to signals that are out of phase from each other by  $\frac{\pi}{4}$  rad. Despite that, LoRa uses the IQ acronym to describe signals with inverted chirp direction, namely up-chirp and down-chirp.

Bacco uses this technique to discriminate between uplink messages (i.e., from Sender to Gateway/Repeater or from Repeater to Gateway) and downlink messages (i.e., from Gateway to Sender/Repeater or 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).

### 2.4.3 Subnetting

The LoRa protocol supports a wide range of carrier frequencies in the VHF spectrum <sup>3</sup>.

Bacco exploits this fact to build sub-networks that operate at different frequencies to achieve very low interference between them. The set of used frequencies is defined by Equation 2.1.

$$[h]f = \{f_k : f_k = 868\text{MHz} + k \times 125\text{KHz}, k \in \{0...10\}\} \quad (2.1)$$

The main sub-network operates at the base frequency of 868.0 MHz, obtained by setting  $k = 0$  in Equation 2.1; it is composed of the Gateway and all the Sender nodes connected to it. Every other sub-network operates at a different frequency that is obtained by choosing  $k \in \{1...10\}$ ; they are composed of a Repeater and all the Sender nodes connected to it. To be able to communicate with the Gateway,

---

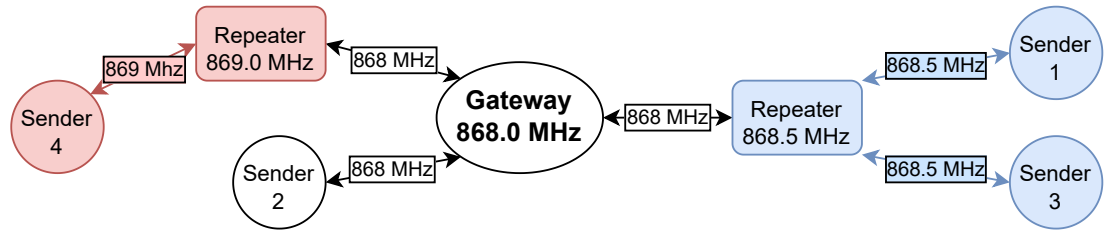
<sup>3</sup>For reference, the SX1262[6] transceiver features a continuous frequency coverage from 150 MHz to 960 MHz



Repeaters need to forward the messages at the base frequency of the Gateway, regardless of their operating frequency inside the sub-net.

A network using Bacco 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. This is very important because of bouncing, a phenomenon that occurs when a message is sent back and forth between Repeaters.

Figure 2.3 shows a network consisting of 3 different sub-networks, represented with different colors; the main one is in white, the blue one has a value of  $k = 2$ , and the red one has a value of  $k = 1$ .



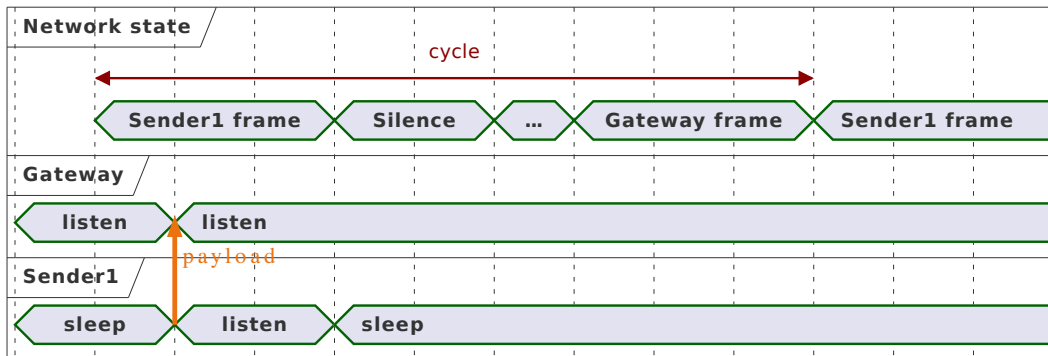
**Figure 2.3:** Example network with subnets in different colors.

#### 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 AlohaNet [7] protocol. Frames are equally distributed between the maximum number of Senders that can be connected to a Gateway (i.e.,  $N_{max} = 254$ ), and the frame assignment is based on the identifier (e.g. Sender1 to slot 1, Sender2 to slot 2, etc.). A Sender can only transmit during its assigned frame, otherwise the Gateway will send a time-correction message. (see Subsection 2.4.4 for a detailed explanation). The time delay between consecutive transmissions from the same Sender is a constant value and it is called  $C$  (cycle time). Between each frame, a time equivalent to  $\frac{1}{3}$  of a Sender frame is left as tolerance and is called radio silence frame. At the end of a cycle, a time equal to  $\frac{C}{5}$  is reserved for the Gateway to upload the collected data. Figure 2.4 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.

**Table 2.1:** Time parameters calculation.

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

**Figure 2.4:** Timing diagram - Sender1 is in sync.

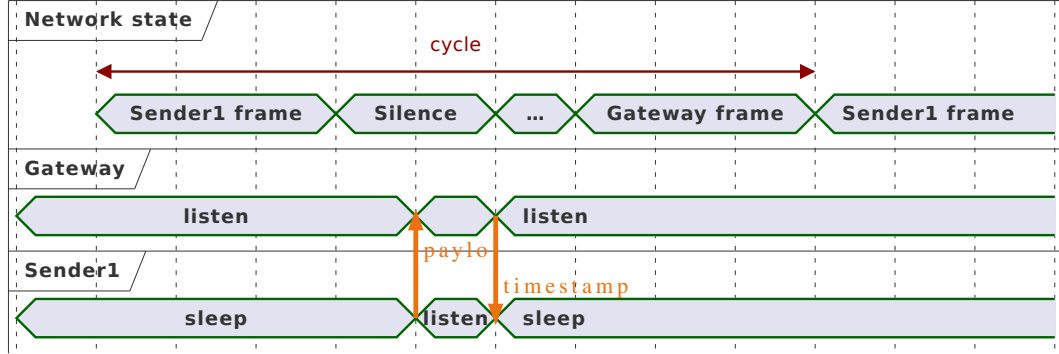
### Clock drift compensation

All the Senders in the network need to transmit during their assigned frame; 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 can not provide a constant frequency source due to manufacturing imprecisions, temperature gradient, etc...

To deal 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. Such a message is sent as soon as the Gateway receives an uplink message that exceeds its correct time frame. Figure 2.5 shows this specific case. In addition to that, the Gateway sends at least 1 downlink message every 10 uplink messages from the same Sender, to indicate that the connection is still active.

## 2.5 Network Discovery

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



**Figure 2.5:** Timing diagram - Sender1 is out of sync.

the most suitable. The Sender tries to establish communication with Repeaters and Gateways throughout all the available frequencies by sending a particular type of message that triggers a sync acknowledge message (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  $\leftarrow$  [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
     $f_k \leftarrow 868 \times 10^6 + k \times 125 \times 10^3$ 
    for  $i$  from 0 to 10 do
      do
        sleep for 1 s
        enter CAD mode at frequency  $f_k$  for 1.5 s
        while activity detected by CAD
          send sensing message
          enter receive mode for 3 s
          if received SYN/ACK then
             $rssi\_values[k] \leftarrow$  current rssi
            break
          end if
        end for
      end for
    end while
  return  $868 \times 10^6 + \text{argmin}(rssi\_values) \times 125 \times 10^3$ 

```

---

## 2.6 Network Joining

When a Sender node needs to connect to the network for the first time, it does not yet have an identifier nor its clock is in sync. The procedure to achieve that will be called the joining process. Note that we assume that the Sender node has already selected the frequency of operation, as described 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 sync message (SYN) to the Gateway and waits for a SYN/ACK response for 3 s. If no message is received, another SYN is sent a maximum of 10 times. After that, the Gateway waits for 3 s for an acknowledge message (ACK) from the Receiver, and if no message is received it will try again for a maximum of 10 times. The SYN/ACK 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 shows a schematic representation of the process.

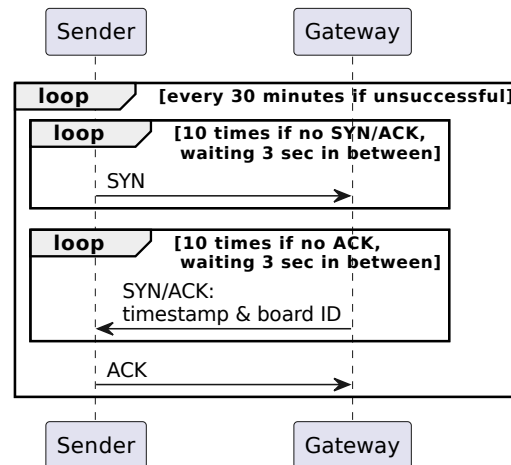


Figure 2.6: Network joining process.

## 2.7 Downlink Commands

In some situations such as transmission power adaption, it is required to be able to change the behavior of the network dynamically and reliably. Bacco achieves

this by exchanging specialized packets that contain commands. Each message contains a command and 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 shows a graphical representation of the process.

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

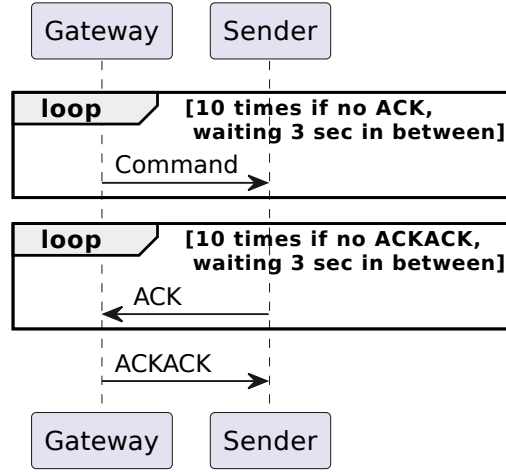
- **SHUTDOWN** - If this command is sent and processed successfully, the Sender goes to deep sleep indefinitely until a manual reset is invoked by pressing a physical button;
- **ENTER SLEEP MODE** - If this command is processed successfully, the Sender stops sending data, but it keeps listening for 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 transmission power by  $P_{step} = 3$  dBm;
- **DECREASE TRANSMISSION POWER** - If this command is processed successfully, the Sender decreases its transmission power by  $P_{step} = 3$  dBm.

**Table 2.2:** Table of opcodes.

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

## 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



**Figure 2.7:** Command sending process.

will automatically drift towards a more suitable value according to the following triggers and Table 2.3:

- If a Sender has not received any downlink messages during the last 20 frames, then its transmission power will be increased by  $P_{step}$  up to a maximum of  $P_{max}$
- If 10 out of the last 10 messages received by a Repeater or a Gateway from the same Sender satisfy  $RSSI > RSSI_{high}$  and  $SNR > SNR_{high}$ , then a command is sent telling to decrease the transmission power by  $P_{step}$  down to a minimum of  $P_{min}$
- If 8 out of the last 10 messages received by a Repeater or a Gateway from the same Sender satisfy  $RSSI < RSSI_{low}$  or have not been received, then a command is sent telling to increase the transmission power by  $P_{step}$ .

**Table 2.3:** Parameters for transmit power adaption algorithm.

Parameter	Value
$P_0 = P_{max}$	14 dBm
$P_{min}$	5 dBm
$P_{step}$	3 dBm
$RSSI_{low}$	-115 dBm
$RSSI_{high}$	-60 dBm
$SNR_{low}$	-7 dBm

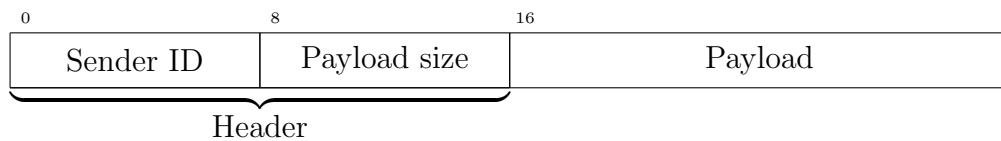
Gateways and Repeaters always operate at  $P_0$ , since power efficiency for sending a message is less critical than ensuring reliable delivery.

## 2.9 Packet Format

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

### Uplink Packet Format

Uplink messages are 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. The least significant byte contains the Sender's identifier represented as an 8-bit unsigned integer. The second least significant byte contains the size of the payload in bytes as an 8-bit unsigned integer. The rest of the message contains the payload and has a length defined by the previous field. Figure 2.8 shows the packet format.



**Figure 2.8:** Uplink packet format.

### Downlink Packet Format

Downlink messages are 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. The least significant byte contains the identifier of the Sender for which the message is directed as an 8-bit unsigned integer. The following bit contains the type of the message: 0 represents a time sync message whereas 1 represents a command message. The content of the following bits depend on the type of message: in the case of a time sync message, the remaining 31 bits contain the timestamp, whereas in the case of a command message, the first 7 bits contain an opcode and the remaining 14 bits are left for padding and can be later used by future revisions of the protocol. 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 for timestamps.



**Figure 2.11:** Downlink packet format for commands.

## Comparison with LoRaWAN

Comparing Bacco's packet format to LoRaWAN's (shown in Figure 2.12 and 2.13, see [8] for more information), we can make some observations:

- Bacco uplink packets are smaller: the firsts have a fixed header size of 2 bytes, while LoRaWAN uses at least 12 bytes for the same purpose;
- Bacco downlink packets are smaller: the firsts have a fixed overall size of 5 bytes, while LoRaWAN uses variable size packets that contain at least a 12 bytes header and can additionally contain a variable-sized payload;
- Bacco has fewer parameters to configure so it is less configurable, but on the other hand it is easier to implement.



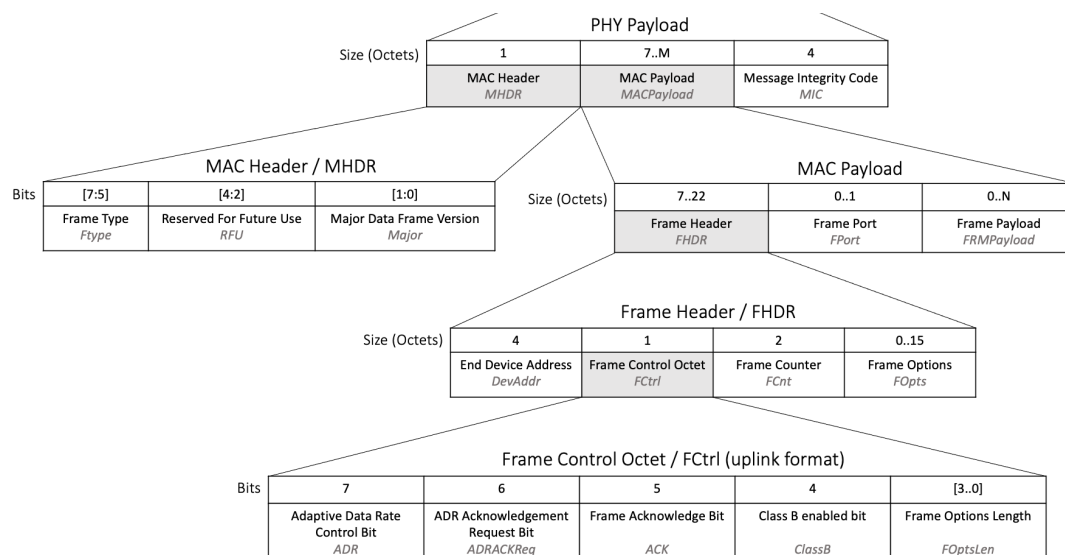


Figure 2.12: LoRaWAN uplink packet format, courtesy of Semtech.

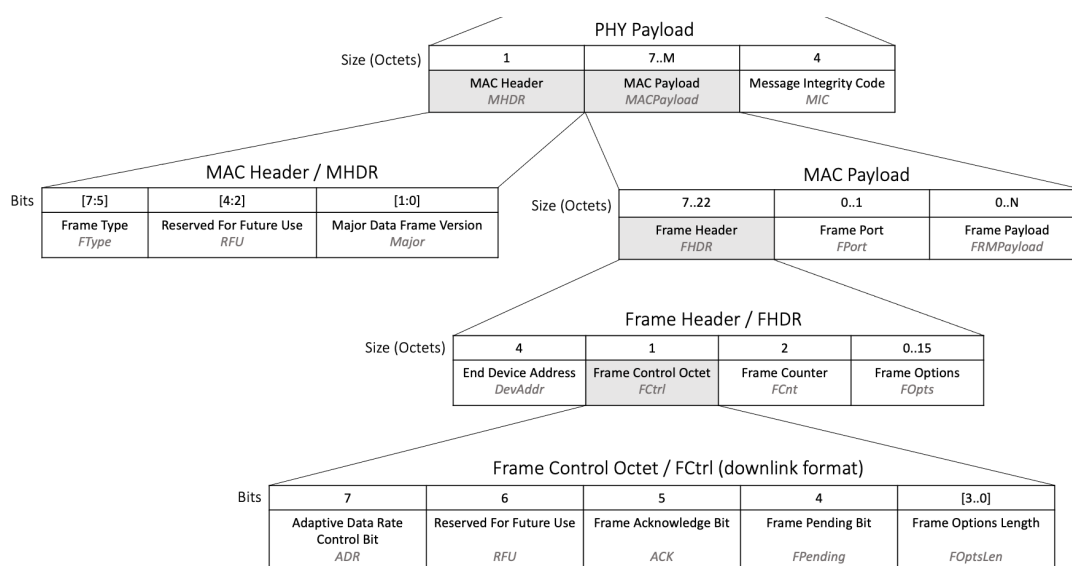


Figure 2.13: LoRaWAN downlink packet format, courtesy of Semtech.



# Chapter 3

## Performance

The goal of this chapter is ...

### 3.1 Time on air

#### 3.1.1 Regulations

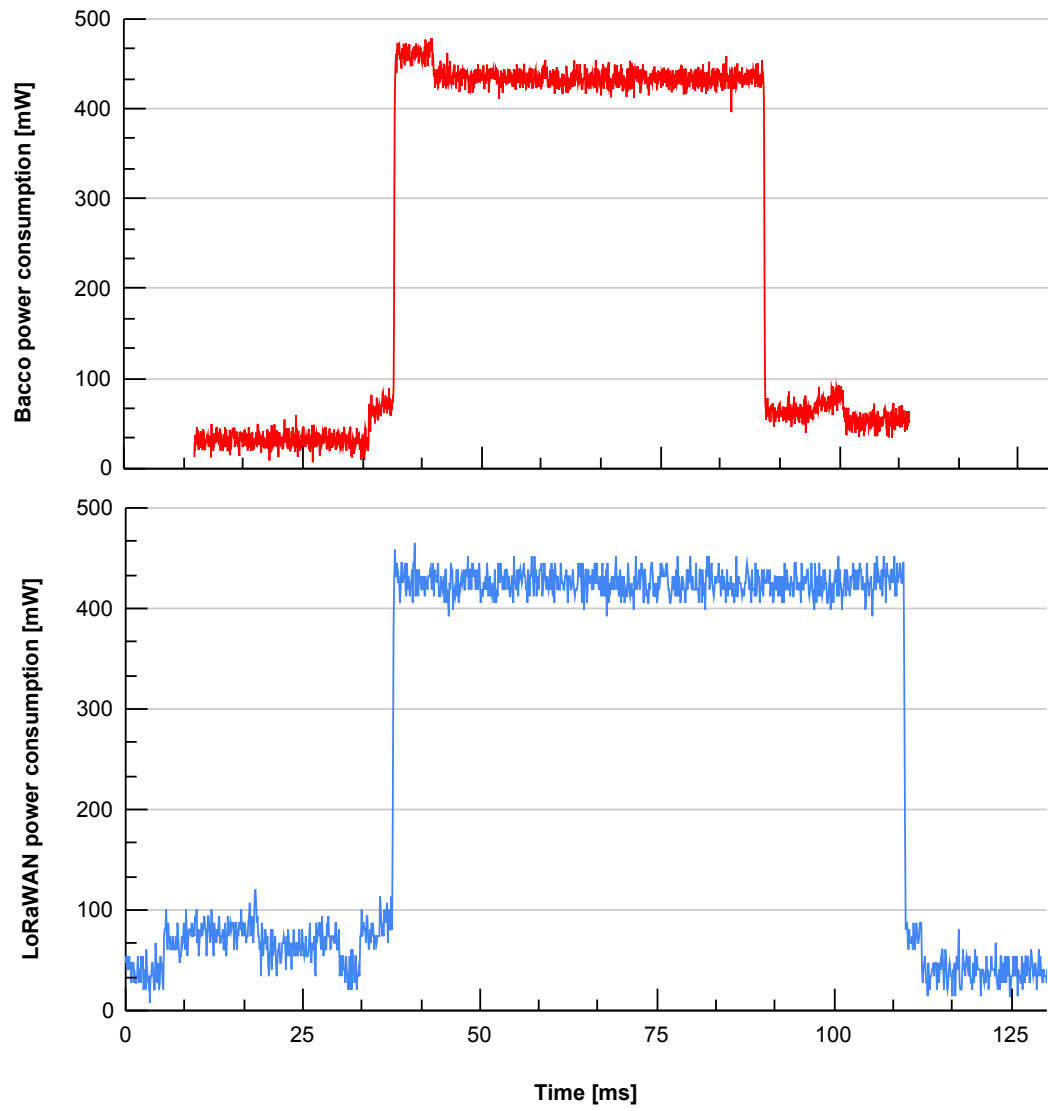
#### 3.1.2 Lab tests

Bacco:

delta time is 51.6ms and total energy is 21.3mJ

LoRaWAN:

delta time is 71.8ms and total energy is 30.8mJ



**Figure 3.1:** Power draw of Bacco (in red) and LoRaWAN (in blue) during the transmission of a packet with a payload of 15 bytes, using SF7, 14dBm, 125kHz bandwidth

# 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](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.fKdAr9ZAo3cJLc4F2cBdUsMftpT2vs0ICP7NmvmMo>.

- [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.
- [8] *In-depth: Sending and receiving messages with lorawan*, Semtech. [Online]. Available: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/sending-and-receiving-messages-with-lorawan/sending-and-receiving-messages/#processing-the-downlink-packet>.