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 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.

TTN The Things Network, an open community of LoRaWAN gateways.

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

Introduction

Place introduction here

Chapter 1

Benefits, Requirements And Use Case Scenario

The goal of this chapter is to describe the the benefits and requirements for internet of things (IoT) in agriculture and and to describe some existing well-knwown solutions. In particular, we will give a brief introduction to the lora wide area network (LoRaWAN) protocol with a focus on some proposed use case scenarios.

1.1 Benefits

The refinment of automation systems used in agriculture has been a costant interest throughout all history of industry due to the huge positive impact it has brought to society. If IoT is introduced in agricultural contexts, it could bring lots of benefits:

- Precision farming: IoT devices enable farmers to collect real-time data on environmental variables. This data helps optimize irrigation, fertilization, and pest control, leading to higher crop yields and reduced resource waste;
- Decision making: the generated data provides farmers with insights into crop health, growth patterns, and yield predictions. This allows them to make informed decisions regarding planting, harvesting, and resource allocation, resulting in better outcomes;
- Remote monitoring and management: farmers can remotely monitor their fields through IoT devices, reducing the need for constant physical presence. This is especially valuable for managing large or distant farms;

- Automation and labor savings: automation can handle tasks such as planting, irrigation, and harvesting. This not only reduces labor costs but also frees up farmers to focus on more strategic aspects of their operations;
- Knowledge sharing and collaboration: IoT platforms can facilitate the exchange of best practices, data, and insights among farmers, researchers, and agricultural experts, promoting knowledge sharing;
- Sustainability and environmental impact: trough optimized resource use, IoT contributes to more sustainable and environmentally friendly agricultural practices;

1.2 Requirements

Where IoT is applied to agriculture, a set of unique and challenging requirements come into play:

- Power management: many remote agricultural locations lack a reliable power source. Devices need to be designed with energy-efficient technology paired with high-energy batteries and/or alternative power sources like solar panels to ensure uninterrupted operation;
- Connectivity and coverage: many agricultural areas lack reliable internet connectivity, which can hinder transmissions between IoT devices and central systems. Even by using alternative solutions such as low power wide area networks (LPWANs) like LoRaWAN or Bacco, long distances and physical barriers are problems that need to be faced;
- Reliability and fail-safeness: human operation is not always convenient or even possible in some cases, so having a fail-safe system is crucial to have maintain functionality;
- Physical damage: constant exposure to weather conditions undermines the integrity of the devices used in the open field, so it is important to use of materials that are resistant to such circumstances;
- Environmental impact: some ecosystems may be influenced by the introduction of alien objects or electromagnetic radiation;
- Cost: initial setup costs for devices, sensors, and infrastructure can be high, which may discourage small farmers from adopting these technologies;

- Regulations and compliance: different regions may have varying regulations concerning the elecromagnetic specturm usage, environmental monitoring, and technology deployment. Adhering to these regulations can be complex when implementing solutions across different areas;

- User-friendly interfaces: the user interfaces need to be intuitive, especially for farmers who might not have extensive technical knowledge.

This thesis will not comprehensively cover all the aspects mentioned above. Instead, it will just concentrate on the communication protocol utilized by the network. While all efforts have been directed towards meeting the requirements, it is important to note that mechanical, environmental, economic, and legal considerations will be left to future discussions.

1.3 An Already Available Technology: LoRaWAN

One of the most promising technologies in IoT for agrigulture is LoRaWAN, an open protocol built on top of Semtech's proprietary LoRa modulation. It provides all the necessary software components to build a suitable network that is reliable, power efficient and scalable according to extensive past research. See [4] and [5] for reference.
A LoRaWAN network consists of sender nodes and gateways. When a sender node broadcasts a LoRa transmission, it can be received by one or more gateways, that can store the information or forward it to a web server through the Internet. This kind of message is called uplink. Gateways might need to transmit data to the sender nodes, this kind message is called downlink. Gateways can be self-owned with a custom configuration or can be part of an existing community such as The Things Network (TTN) or Helium.
Senders can uplink at any time because gateways are always listening to transmissions. On the other hand, downlink messages need to be scheduled as it is not always possible for a sender node to be listening continuously because of the limited power budget. To achieve that, LoRaWAN defines 3 classes of devices called class A, class B and class C. The classes are sorted by increasing energy demand, so class A stands out in agricultural contexts thanks to its superior power efficiency when compared to classes B and C. In this mode, a downlink message can only be sent in 2 time slots at pre-defined delays subsequently to the reception of an uplink message. Figure 1.1 shows a schematic version of the time

slot management.
Due to the precise time slot management, the LoRaWAN standard does not support repeaters. Successful research has been done to overcome this limitation, see [6], however the additional required harware and the lack of official support from the standard makes it unconvenient to use.

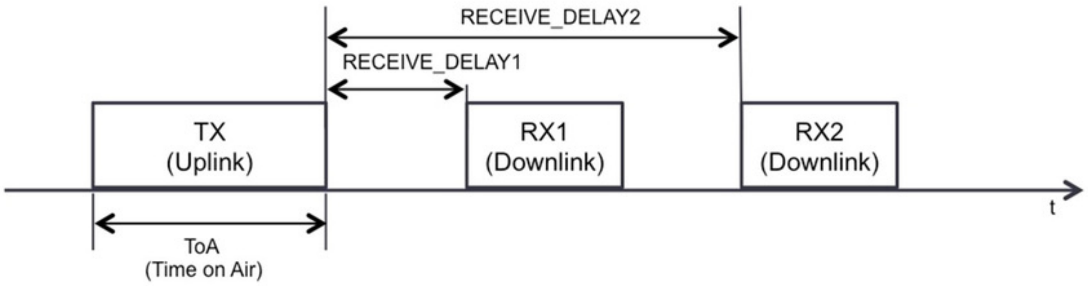

Figure 1.1: LoRaWAN class A device operation.

1.4 Real Word Use Case Scenario

To better understand the limitations and the strenghts of LoRaWAN and Bacco, I will now describe 3 real world scenarios in which IoT can be integrated to achive the benefits described in Section 1.1. All of them require to collect physical data on a vineyard, however the locations and the surface areas are different for the 3 cases. Here follows a brief description of each of them:

1. A small vineyard with an area of 1 hectar [1] which is located near an urban area and thus is fully covered by one or more hostpots registered on a public LoRaWAN network. The surface of the terrain is almost flat;

2. A small vineyard with an area of 1 hectar which is located far from any urban area and thus is not fully covered by any public hostpot. The surface of the terrain is almost flat;

3. A large vineyard with an area of 100 hectars which is located far from any urban area and thus is not covered by any public hostpot. The vineyard is spread across multiple hills.

[1] 1 hectar is equivalent to 10^4 m^2

1.4.1 LoRaWAN Setup Using A Public Gateway

We will now analyze an IoT network that makes use of a public infrastructure such as TTN or Helium. Each sender node can be easily registered on the platform and can transmit independently from the other devices.
When trying to apply this solution to each of the described scenarios, we would observe the following behavior:

1. Since the vineyard is covered by one or more public hotspots, it is possible to build the desired system. Depending to the distance from the hotspot, the sender nodes may be required to transmit at a high power in order for the signal to be detected by the hotspot. Any fault of nearby hotspots will result in an unusable and not directly fixable system;

2. Since the vineyard is not fully covered by any public LoRaWAN network, it is not possible to build the desired system;

3. Since the vineyard is not fully covered by any public LoRaWAN network, it is not possible to build the desired system.

This is the simplest and cheapest proposed solution to implement, as no additional hardware is needed other than the sender nodes, however it does not satisfy the requirements for 2 out of the 3 proposed scenarios.

1.4.2 LoRaWAN Setup Using An Owned Gateway

We will now analyze an IoT network that makes use of owned/private LoRaWAN hotspots. Each sender node can be connected to it and can transmit independently from the other devices.
When trying to apply this solution to each of the described scenarios, we would observe the following behavior:

1. Since the vineyard is small and the underlying surface is mostly flat, the hotspot can fully cover it. The system will satisfy the requirements;

2. Since the vineyard is small and the underlying surface is mostly flat, the hotspot can fully cover it. The system will satisfy the requirements;

3. Since the vineyard is large and spread across multiple hills, it may not be possible for a single hotspot to cover the entirety of it because of physical barriers, so multiple hotspots would be needed to satify the requirements.

The solution satisfies the requirements for all the scenarios, however in the 3^{rd} case it is necessary to use multiple gateways to achieve full coverage. This makes the system very expencive and difficult to deploy.

1.4.3 Bacco Setup

Bacco is described in this thesis as an alternative to the discussed existing solutions. It has the goal to satisfy all the requirements for the proposed scenarios and to provide a competitive system to LoRaWAN in the field of agriculture. The next chapters will describe its implementation details and compare its performance to LoRaWAN's.

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

We will now describe a simple network that makes use of Bacco to better understand its operating principle. The network is built upon 4 categories of devices:

- Sender node - collects data and sends it to a Gateway or Repeater using Bacco over LoRa modulation;
- Repeater node - listens to incoming Bacco messages from Senders and forwards them to a Gateway;
- Gateway node - collects data coming from Sender or Repeater nodes and sends it to the web server. 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. This node has the role of coordinating and synchronizing Sender nodes. It can be optionally configured to perform pre-processing operations (e.g. filtering, smoothing, interpolation ...) on the incoming data;
- Web server - receives data coming from the Gateways, elaborates it, and makes it available to the user. Note that the scheme of communication involving this device is not covered by Bacco.

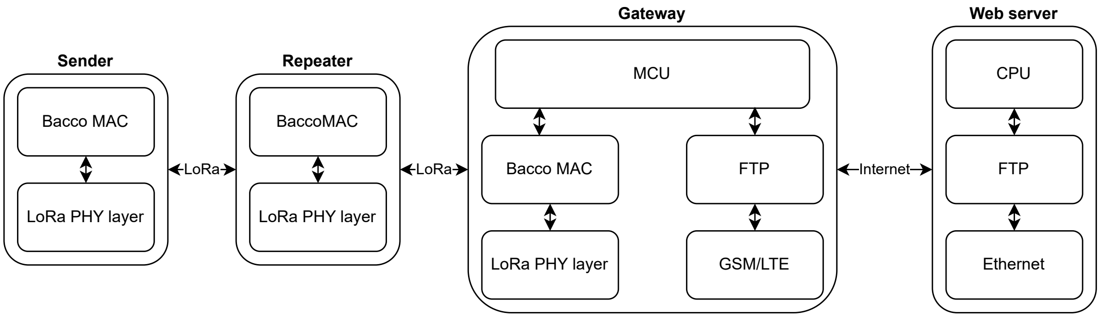

Figure 2.1: Schematic representation of an example network using Bacco.

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.

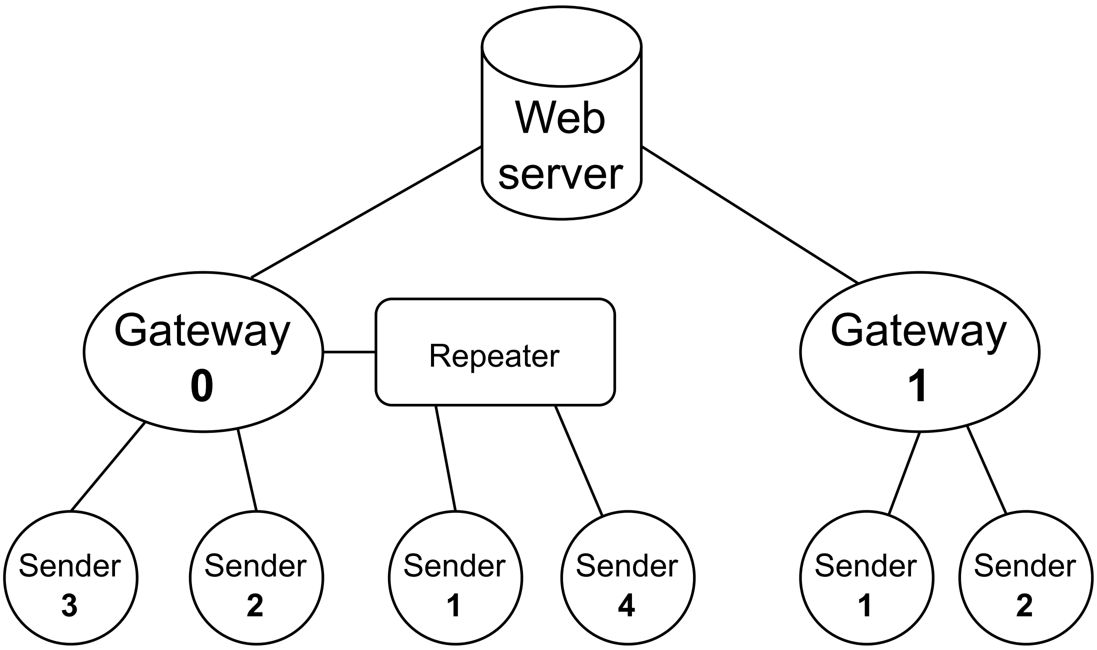

Figure 2.2: Example network topology

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 number of Sender nodes connected to a single Gateway to 254 [1]. 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 transcievers [8]. In this mode, the LoRa node listens for any transmission on a specific 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

[1] 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** [7] on duty cycle for the 868MHz band and the fact that most agricultural contexts do not require a huge amount of sensors

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 very high frequency (VHF) spectrum [2].
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.

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

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, 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 occours 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 $k = 1$.

[2]For reference, the SX1262[9] transceiver features a continuous frequency coverage from 150 MHz to 960 MHz

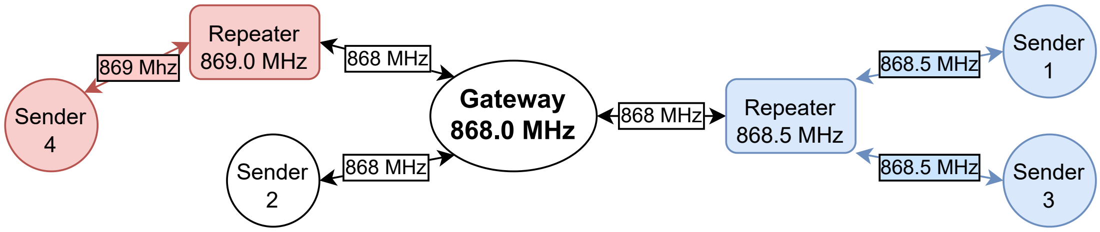

Figure 2.3: Example network with subnets in different colors.

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 [10] 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	$\frac{0.2 \cdot C}{N_{max}}$

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

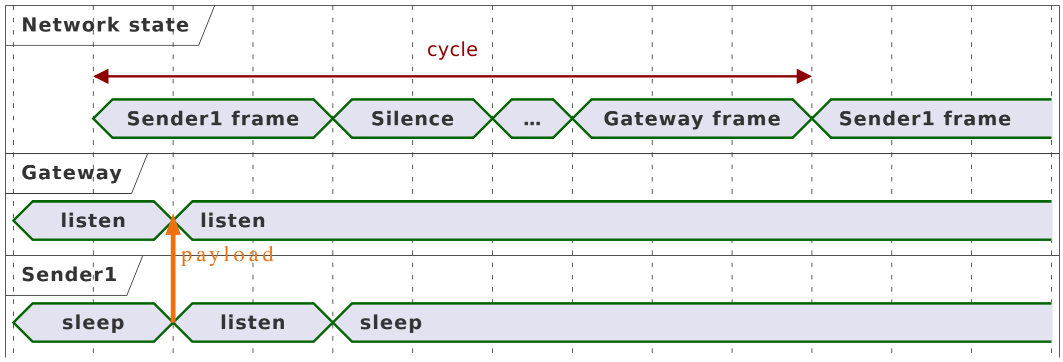

Figure 2.4: Timing diagram - Sender1 is in sync.

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.

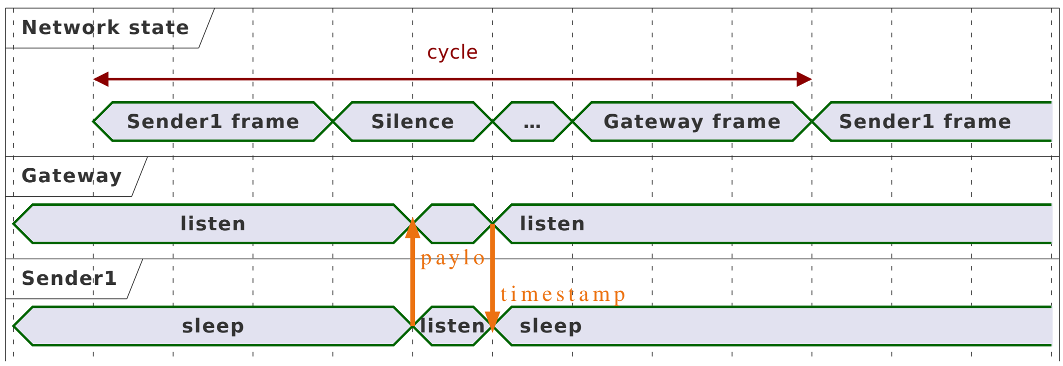

Figure 2.5: Timing diagram - Sender1 is out of sync.

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 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 can not 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 ← 868 × 10^6 + k × 125 * 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] ← current rssi
                break
            end if
        end for
    end for
end while
return 868 × 10^6 + argmin(rssi_values) × 125 * 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.

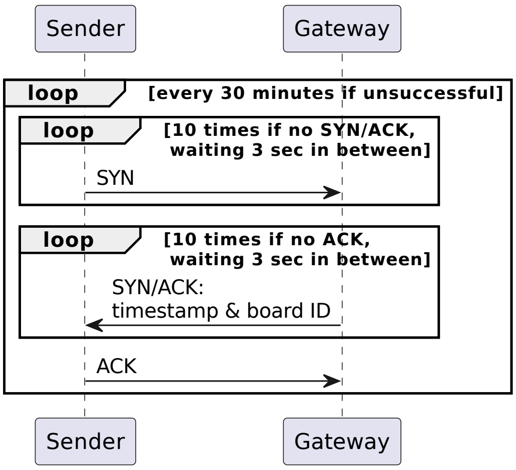

Figure 2.6: Network joining process.

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]

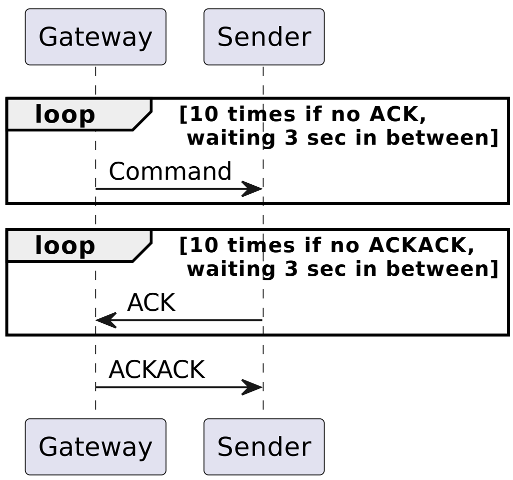

Figure 2.7: Command sending process.

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 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.

2.9.1 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.

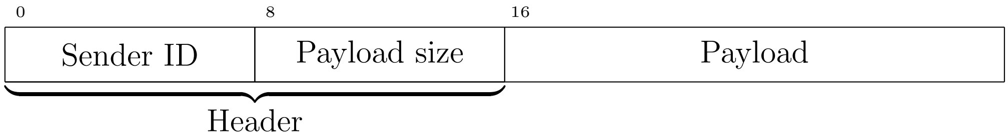

Figure 2.8: Uplink packet format.

2.9.2 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.

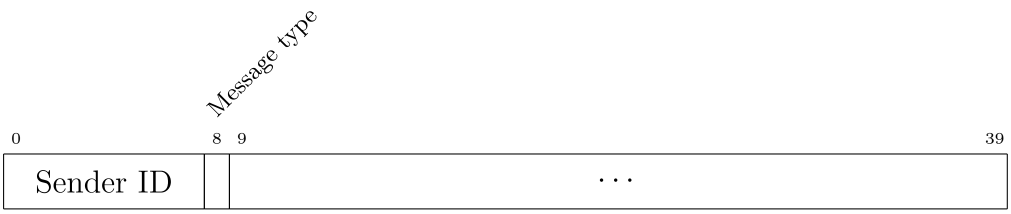

Figure 2.9: Downlink packet format.

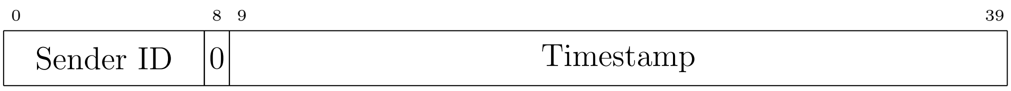

Figure 2.10: Downlink packet format for timestamps.

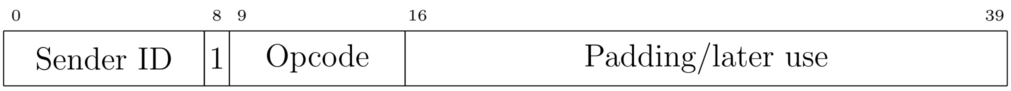

Figure 2.11: Downlink packet format for commands.

2.9.3 Comparing Bacco And LoRaWAN Packet Formats

We will now compare Bacco's packet format to LoRaWAN's. The latter is described thoroughly by Semtech's official documentation [11], a schematic reference

can be found in Figure 2.12 and 2.13.
The focus of the comparison will be on the total size of the packets rather than the function of each individual field, in order to analyze the overhead associated to each protocol.

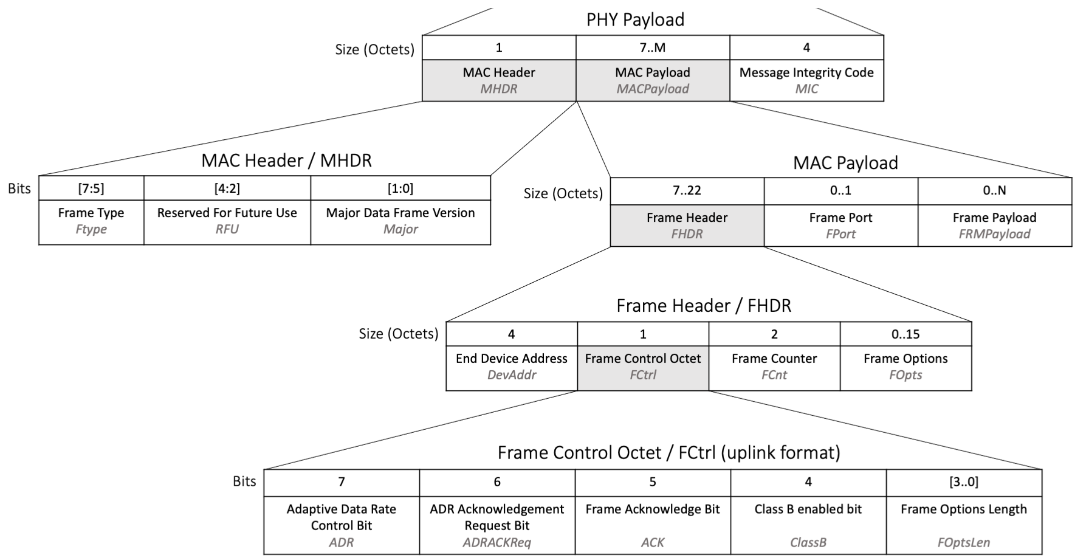

Figure 2.12: LoRaWAN uplink packet format.

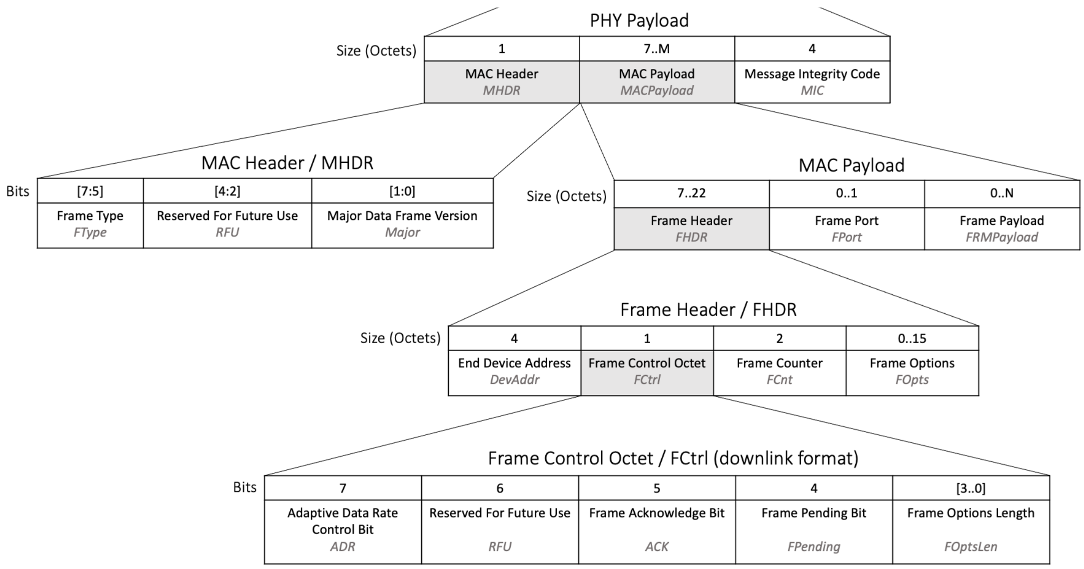

Figure 2.13: LoRaWAN downlink packet format.

Uplink Packets

Considering two uplink packets containing the same payload and trasmitted with the two protocols under analysis, we can observe that Bacco uplink packets feature

a fixed header size of 2 bytes as described in Section 2.9.1, while LoRaWAN uses a header size that varies between 12 and 28 bytes.
To calculare these numbers we can consider an empty *Frame Paylaod* and sum all the other fields' sizes to get the overall header size. We first consider the lower bound for each field to get a total of 12 bytes (MAC Header: 1 + Frame Header: 7 + Frame Port: 0 + Message Integrity Code: 4) and the the upper bound to get a total of 28 bytes (MAC Header: 1 + Frame Header: 22 + Frame Port: 1 + Message Integrity Code: 4). This shows that Bacco uplink packets are always shorter at a fixed paylaod size.

Downlink Packets

Considering two uplink packets containing the same payload and trasmitted with the two protocols under analysis, we can observe that Bacco downlink packets feature a fixed overall size of 5 bytes as discussed in Section 2.9.2, while LoRaWAN uses packets that contain a header with a size between 12 and 28 bytes (the calculation is analogous to the one of the uplink case, discussed in Section 2.9.3). This shows that Bacco downlink packets are always shorter at a fixed paylaod size.

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

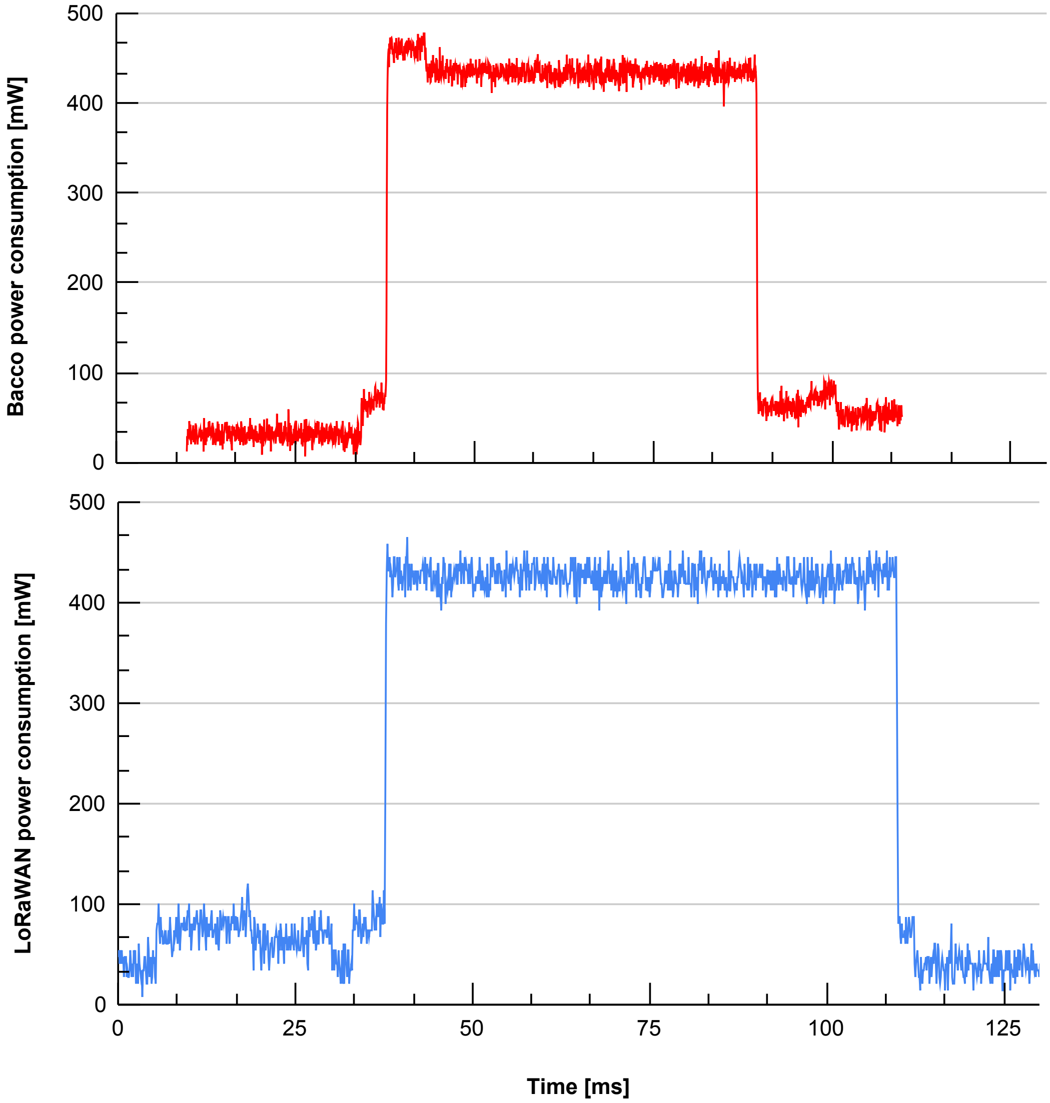

Figure 3.1: Power draw of Bacco (in red) and LoRaWAN (in blue) during the transmission of a packet with a payload of 15 bytes, using SF7, 14dBm, 125kHz bandwidth

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] D. Davcev, K. Mitreski, S. Trajkovic, V. Nikolovski, and N. Koteli, "Iot agriculture system based on lorawan", in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2018. DOI: 10.1109/WFCS.2018.8402368.

[5] P. Boonyopakorn and T. Thongna, "Environment monitoring system through lorawan for smart agriculture", in *2020 - 5th International Conference on Information Technology (InCIT)*, 2020. DOI: 10.1109/InCIT50588.2020.9310977.

[6] E. Sisinni, P. Ferrari, D. Fernandes Carvalho, *et al.*, "Lorawan range extender for industrial iot", *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5607–5616, 2020. DOI: 10.1109/TII.2019.2958620.

[7] *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.

[8] *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.

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

[10] 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.

[11] *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.