LoRa/LoRaWAN

Desportes Kilian, Hadouch Ilhame, Imekraz Yanis, Mahraye Abderrazzak

Introduction

Several protocols of communication take place in IoT. In this report, we will study the protocol LoRaWAN and the technology LoRa. LoRa (Long Range) is a low-power wide-area network (LPWAN) protocol that enables long-range transmissions (more than 10km in rural areas) with low power consumption. First, we will focus on the physical and MAC layers, then we will show why LoRa has a low energy consumption. We will end with a discussion about the security in this network.

Physical layer – frequency, bandwidth, modulation, range

Frequency, range, and bitrate:

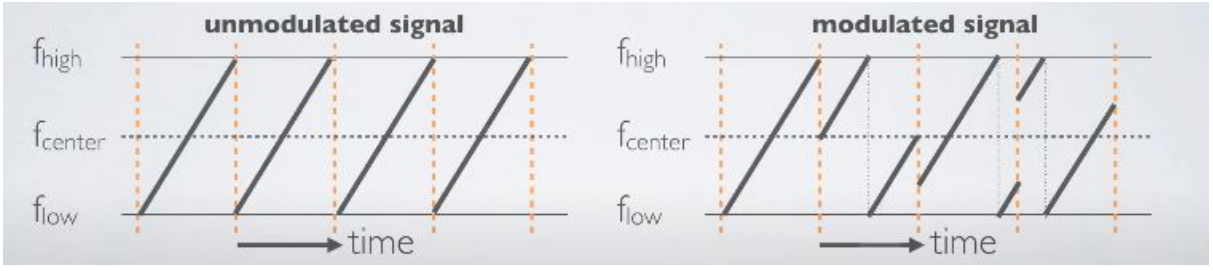
LoRa allows long range (up to 15kms) and low power communication. It can use the 433-, 868- or 915-MHz bands depending on the deployment location (868Mhz in Europe).

The LoRa's payload can reach up to 255 octets, with a bitrate which can reach the 50Kbps. [1]

Modulation:

The LoRa protocol is based on a chirp spread spectrum (CSS) modulation. CSS is a spread spectrum technique that uses wideband frequency to encode information. This modulation is proprietary and is not open.

Here is an example of this modulation: The data is encoded into chirps thanks to frequency jumps.



Spreading factor:

One of the most important parameters in LoRa's physical layer is the spreading factor. The Spreading Factor (SF) decides on how many chirps, the carrier of the data, are sent per second. In fact, the lower is the SF the more chirps can be sent per second and then, the more data can be encoded per second.

Consequently, this spreading factor acts on the range, and the bite rate. In fact, the higher is the SF the further can be the range. And, the higher is the SF the lower the bitrate can. This can be explained by this equation [2]:

$$R_b = SF \times \frac{BW}{2^{SF}} \times CR \quad (1)$$

MAC layer – LoRaWAN

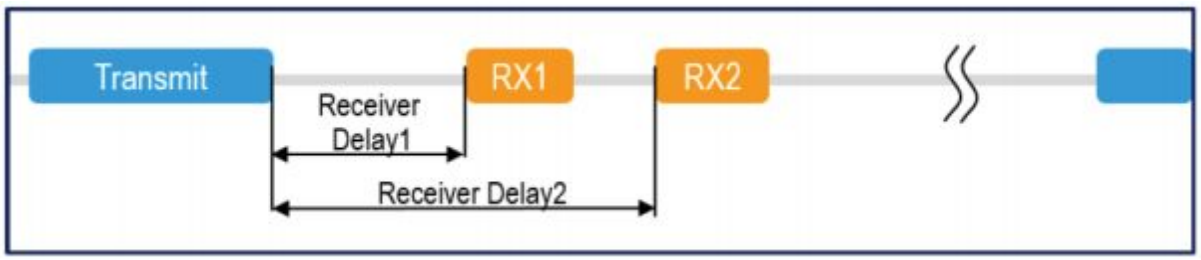
LoRaWan Protocol :

LoRaWAN is a MAC protocol. It is asymmetric, using uplink and downlink messages.
This protocol is opened to everyone. It is used both on connected objects (end devices) and gateways/servers. The communication betweens these is half duplex, using uplink messages (end devices to gateways) and downlink messages (gateways to end devices).

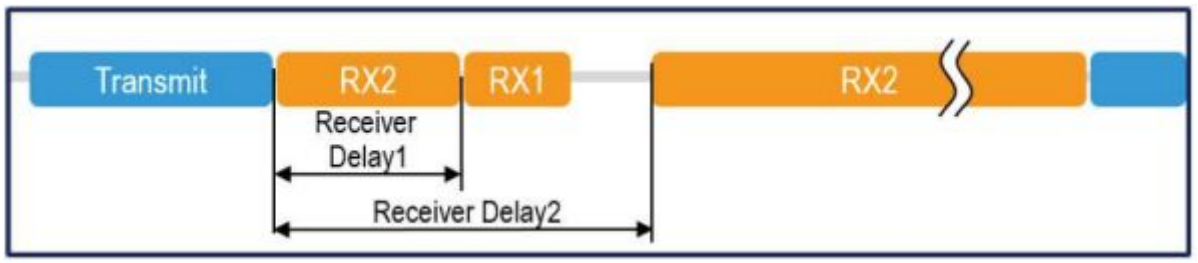
How it works :

Initially, gateways are in listening mode. When an object transmits data, every gateway receives it and transmits it to the server with added information, an RSSI (received signal strength indication). When the server responds, it only sends the response to the gateway that gives the best RSSI.
There are 3 classes of transmission, each consuming different levels of energy.

- Class A : The end device transmits a message. The server has two temporal windows during which he can transmit data to the end devices, through the best gateway.



- Class B : It works the same as class A, but the server can transmit a Beacon synchronisation frame that will allow him to get additional transmission frames during 128s.



- Class C : The end device is constantly in reception mode, except when it is sending data. The server can, with this class, transmit at any time, without delay. This class is really effective due to the lack of delay, but is also really energy consuming.

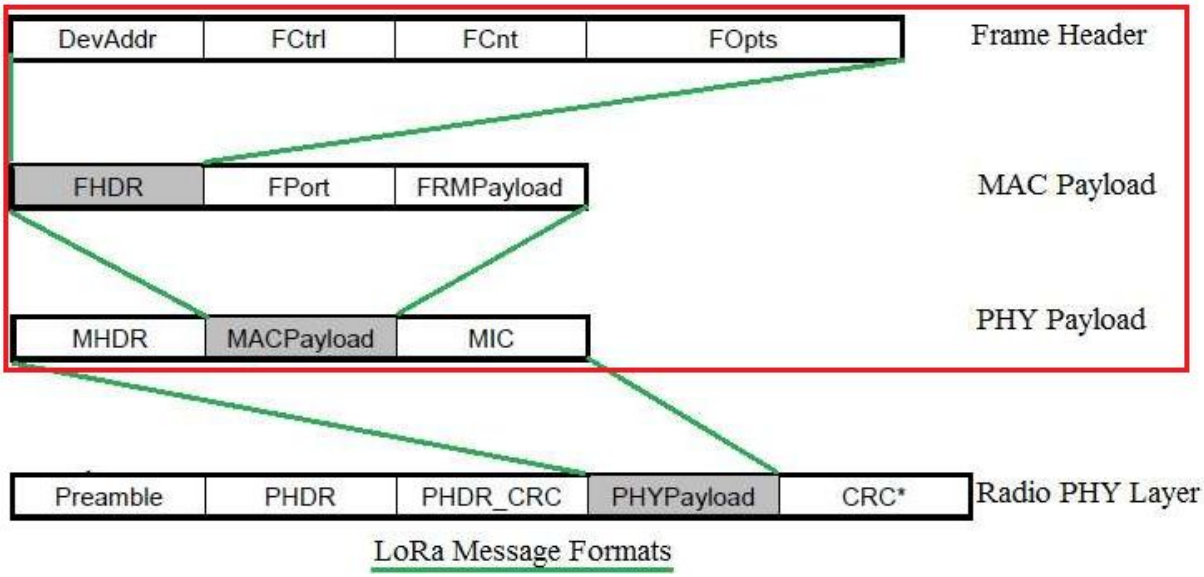
MAC Commands :

These commands are used for **network administration** between server/gateway and end device. They are non visible for applications.

Single dataframe = multiple MAC commands.

All MAC commands are visible here :
https://www.rfwireless-world.com/Tutorials/LoRaWAN-MAC-layer-inside.html

LoRaWAN Medium Access Control (MAC) message format :



LoRa Message Formats

Mac header (MHDR) :

- MType :
  - Join Request & Join Accept: These messages are used to establish connection between LoRa end device and Gateway.
  - Confirmed Data Message: Message that requires to be acknowledged.
  - Unconfirmed Data Message: Message that does not require to be acknowledged.

    - Proprietary : For non standard message format functionalities.
- RFU
- Major

MIC - Message Integrity Code (defined by each message type)

FHDR - Frame Header (Containing address DevAddr, frame control FCtrl, frame counter FCnt and frame options FOtps)

FPort - Port Field

FRMPayload - MAC Frame Payload Encryption (Can contains multiple MAC Commands)

LoRa - Energy consumption

When we need to make a choice of which technology to use for an IoT project, energy consumption is an important parameter to take in consideration. Since the beginning of LoRa and LoRaWAN, some parameters have been studied such as time transmission, range, rate of flow and network capacity. Several parameters are used to study the energy consumption which is optimized depending on the used mode (emission, reception, sleep and wait), the transmission, the power emission… [9] In this part, we will talk about a few of them.

If we consider the ALOHA method, a star topology, the nodes must be available to synchronize with a mesh network or with a synchronous network. With LoRaWAN, we have asynchronous events, devices transmit when they have data available to send. This means that, compared to protocols using synchronization as Wi-Fi, LoRaWAN consumes less energy and increases lifetime battery. The next figure presents the power consumption of different protocols in function of the range coverage. It shows that LoRaWAN has a big advantage because it covers a long range with less power consumption. This is what we can see in the figure (a). [10]

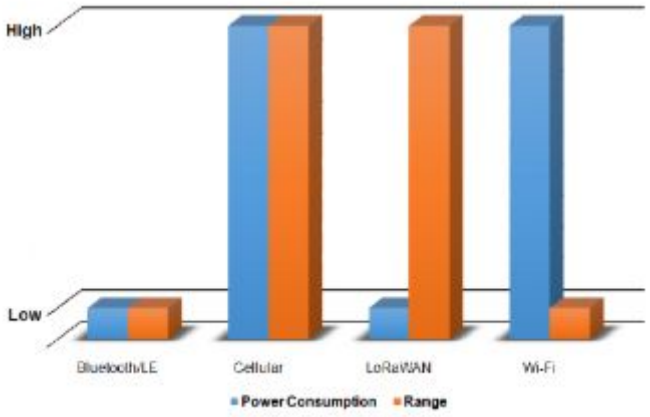


Fig.a: Power Consumption vs Range for Bluetooth/LE, Cellular, LoRaWAN and Wi-Fi technologies

Now, we are going to study the influence of some parameters on energy consumption. Here we take the energy model used in the document [11] for a communicating sensor. The study is based on different equations that we can admit in order to understand the energy consumption/ the power consumption:

- The total consumed energy: $E_{Total} = E_{Sleep} + E_{Active}$ **(2)**

In this study, we take in account the sleep mode because the duration is long.

- The dissipated energy by the node in the sleep mode: $E_{Sleep} = P_{Sleep} \cdot T_{Sleep}$ **(3)**

where P_{Sleep} is the power consumption and and T_{Sleep} the time duration in the sleep mode. We can say that E_{Sleep} evolves proportionally to P_{Sleep} and T_{Sleep}.

- The total energy consumption during the active mode of the microcontroller:

$$E_{Active} = E_{WU} + E_m + E_{proc} + E_{WUT} + E_{Tr} + E_R \quad \textbf{(4)}$$

where E_{WU}, E_m, E_{proc}, E_{WUT}, E_{Tr} and E_R are, respectively, the consumed energies in the system wake-up, the data measurement, the microcontroller processing, the wake-up of the LoRa transceiver, the transmission mode and the reception mode.

- E_{Tr} depends on T_{Tr} which is its time duration expressed as : $T_{Tr} = N_{bit} \cdot T_{bit}$ **(5)**

Equation (5) allows us to conclude that the energy consumption depends on the number of transmitted bits (N_{bit}) and the duration of one bit transmission (T_{bit}).

- The energy per useful bit: $E_{bit} = E_{Total} / (8 \cdot PL) = P_{cons}(P_{Tr}) \cdot T_{Packet} / (8 \cdot PL)$ **(6)**

where PL, E_{Total} and $P_{cons}(P_{Tr})$ are, respectively, the payload size, the total consumed energy and the total consumed power which depends on transmission power. The total consumed power and the energy per useful bit are proportional.

Also, in the first part of this report, we have talked about the SF and this coefficient is a parameter to take in consideration for the energy consumption analysis. With the previous model used, we also have this equation:

$$E_{bit} = P_{cons}(P_{Tr}) \cdot (N_{Payload} + N_P + 4.25) \cdot 2^{SF} / (8 \cdot PL \cdot BW) \quad \textbf{(7)}$$

with $N_{Payload}$ the number of payload symbols and N_P the preamble symbol number.

Equation (7) shows that the bandwidth (BW) , the payload size (PL) and the spreading factor (SF) have an effect on the energy per useful bit. We can see this effect on the following graphs.

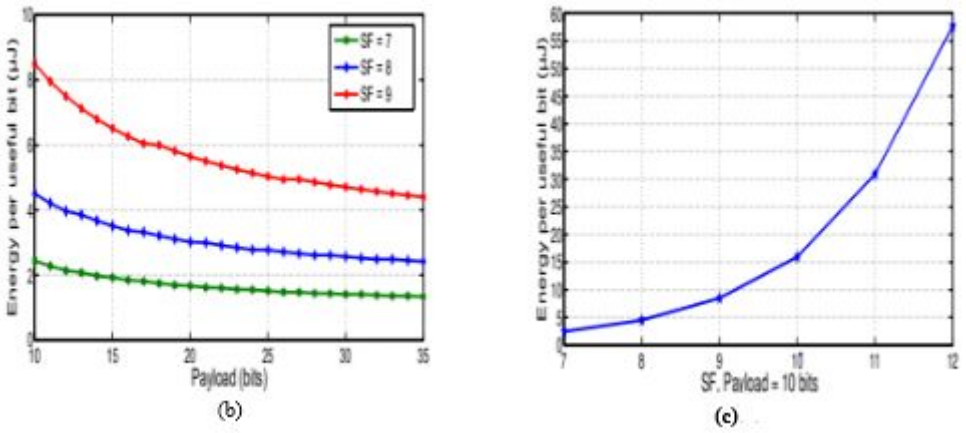


Fig (b): Effect of SF on the consumed energy, CR = 4/5 ; and (c): Energy per useful bit evolution as a function of SF

The consumed energy per useful bit as a function of the payload for different SF is presented in figure (b). This energy decreases with the increase of the number of useful bits and Equation (7) allows us to understand this phenomenon. This result is shown in figure (c) which depicts the evolution of the energy per useful bit as a function of SF for constant payload size (equal to 10 bits). We can conclude that the greater value of SF, the more time is taken to send a packet, so the more consumed energy is needed to transmit data. [11]

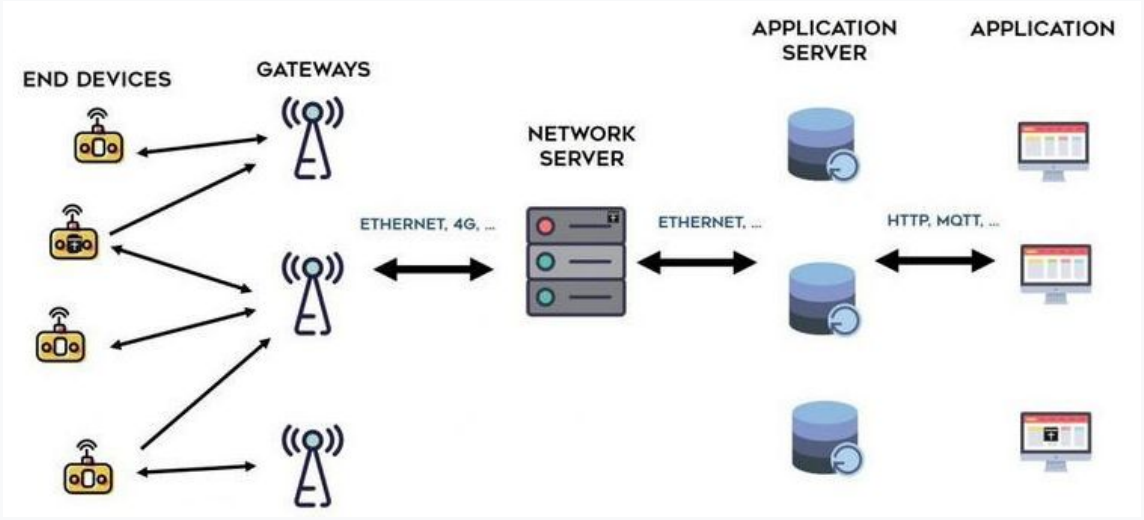
As we have seen in the equation (1), the coding rate (CR) is an important element. Figure (d) shows the effect of the CR on the energy per useful bit. When the coding rate decreases, the time on air and the consumed energy and the total consumed power (because of the proportionality seen in equation (6)) are increased.

Fig (d): Effect of CR on the consumed energy, SF = 7 and BW = 500 KHz

In order to reduce the consumed energy by the sensor node, we have seen that there are several LoRa parameters we can optimize such as SF, CR and payload size.

Lora – Security

Before starting with the security aspect of LoRa, we will review the architecture of the LoRa technology :



The architecture of a LoRaWAN network is composed of several end-devices that are connected with at least one gateway with a single hop. Those gateways will forward the packets they received to the Network Server (NS) through a back-haul network which used IP protocols. The NS is responsible for verifying addresses of end-devices, checking received frames authenticity and frame counters,transferring the payload from and to application servers and more…

The end devices or nodes, to take part in a LoRaWAN network, have to be activated. There is two ways for a node to be activated :

- Over-The-Air Activation (OTAA): Before activation, each node has a 128bits AppKey, and a 64bits DevEUI and AppEUI (EUI stands for Extended Unique Identifier). The DevEUI uniquely identifies the end-device and is similar to a MAC address while the AppEUI uniquely identifies the application server and is similar to a port number. The AppKey is a 128 bit key which is specific to the end device. It is used to generate the MIC (Message Integrity Code) to ensure the integrity of the message.

  On the server side, it must also know the same AppKey as the device.

  The end device will generate an DevNonce which is a randomly generated number. By doing so it will prevent other devices replaying the join request. The end device builds a message containing the devNonce, AppEUI and DevEUI. Over this message the MIC is generated using the AppKey as such :

```
mac=aes128_cmac(AppKey, MHDR | AppEUI | DevEUI | DevNonce)
MIC = mac[0..3]
```

  The end device then sends to the server a join-request containing the DevNonce, the AppEUI, the DevEUI and the MIC.

  Once the Network Server receives the join-request, it'll check if the DevNonce has been used before and then recalculate the MIC using the AppKey stored. If both MIC are the same then the end device is authenticated. The Network Server will then generate it's own AppNonce and calculate the AppSKey (Application session key) and NwkSKey (Network SessionKey).Those are the two new 128bits key of the end device.

```
NwkSKey = aes128_encrypt(AppKey, 0x01 | AppNonce | NetID | DevNonce | pad16)
AppSKey = aes128_encrypt(AppKey, 0x02 | AppNonce | NetID | DevNonce | pad16)
```

  A MIC is generated based on the AppKey of the end device. A message containing the DevAddr, AppNonce, NetID, RxDelay (configuration Data) and CFList (Channels to use) is built and the Network Server sends the response back to the end device containing the encrypted message with the AppKey and the MIC.

  The end device can now calculate it's session keys : AppSKey and NwkSKey.

  The NwkSKey is used by the end device and the Network Server to calculate and verify the message integrity code (MIC) to ensure data integrity. The NwkSKey is also used to encrypt and decrypt the payload. The application session key is used to secure end-to-end communications between the end device and the Application Server. The shared symmetric key is used by the application server and end device to encrypt and decrypt the payload. The payload is end to end encrypted between the end device and the application server but they are not integrity protected. That

means a network server may be able to alter the content of the data messages in transit but the network servers are considered as trusted.

- Activation by Personalisation (ABP) : In ABP mode there are no Join-request nor Accept-message send. The end device already has the keys they need to communicate.

Once an end device is part of a LoRa network, all the messages they will exchange will be encrypted using a combination of NwkSKEy and AppSKey. The FPort is set to 0 if the packet destinator is a Network Server. In this case the NwkSKey is used. Otherwise, the AppSKey is used and the destinator is an Application Server. The encryption of messages is performed using AES128 in Counter mode which is a symmetric key encryption method.

References

- Physical Layer

[1] https://hal.archives-ouvertes.fr/hal-02289990/document
[2] https://www.mdpi.com/1424-8220/16/9/1466/htm

- MAC Layer

[3] https://www.sciencedirect.com/science/article/pii/S1877050918305283
[4] https://www.rfwireless-world.com/Tutorials/LoRaWAN-classes.html
[5] https://www.rfwireless-world.com/Tutorials/LoRaWAN-MAC-layer-inside.html
[6] https://www.rfwireless-world.com/Tutorials/LoRa-protocol-stack.html
[7] https://hal.archives-ouvertes.fr/tel-02415918/document
[8] https://hackmd.io/@hVCY-lCeTGeM0rEcouirxQ/S1kg6Ymo-?type=view

- Energy consumption

[9] https://hal.archives-ouvertes.fr/tel-02092386/document
[10] https://www.researchgate.net/publication/318866065_LoRaWAN_-_A_Low_Power_WAN_Protocol_for_Internet_of_Things_a_Review_and_Opportunities
[11] https://www.mdpi.com/1424-8220/18/7/2104/htm
[12] https://www.researchgate.net/publication/301597433_A_Study_of_Efficient_Power_Consumption_Wireless_Communication_Techniques_Modules_for_Internet_of_Things_IoT_Applications

- Security

[13] https://labs.f-secure.com/assets/BlogFiles/mwri-LoRa-security-guide-1.2-2016-03-22.pdf
[14] Analysis of LoRaWAN v1.1 Security, Ismail Butun, Nuno Pereira, Mikael Gidlund, June 2018