

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Grado

**Provisión de servicios de la Internet
de las Cosas sobre redes basadas en
LoRa**

**(Internet of the Things service provision on
LoRa networks)**

Para acceder al Título de

***Graduado en
Ingeniería de Tecnologías de Telecomunicación***

Autor: Nicolás Alonso

Octubre 2018



E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Nicolás Alonso Villaverde

Director del TFG: Jorge Lanza Calderón

**Título: “Provisión de servicios de la Internet de las Cosas sobre redes
basadas en LoRa”**

Title: “Internet of the Things service provision on LoRa networks”

Presentado a examen el día:

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:

Presidente (Apellidos, Nombre): D. Ramón Agüero Calvo

Secretario (Apellidos, Nombre): D. Alberto Eloy García Gutiérrez

Vocal (Apellidos, Nombre): D. Jorge Lanza Calderón

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Resumen

En una etapa tecnológica en efervescencia, surge el concepto de Internet de las Cosas (IoT), donde se consigue una completa interacción entre personas y su entorno a través del uso de las redes inalámbricas, dando al usuario un mayor conocimiento sobre aquello que le rodea.

En este ámbito el tipo de despliegues que se han popularizado versa desde tecnologías de bajo alcance, el uso de protocolos tradicionales como WiFi o la opción con un mayor desarrollo actualmente; nuevas tecnologías, libres, con amplio alcance donde a día de hoy se encuentra la batalla por decidir cuál es más adecuada y viable.

Dentro de esta última opción surge el protocolo LoRa y su estándar complementario LoRaWAN con el objetivo de implementar despliegues IoT que cubran grandes distancias con bajo consumo. Estos dos protocolos que cumplen sendos requisitos, ofrecen al usuario flexibilidad respecto al modo en que se diseñen los sistemas, así como múltiples comunidades que dan soporte a sus despliegues.

Este proyecto nace con el propósito de profundizar en la tecnología LoRa y LoRaWAN tratando de validar si son funcionales, así como identificar sus puntos fuertes y debilidades.

Para conseguirlo, se plantean dos modos de actuación a lo largo del proyecto. Por un lado, la caracterización del canal radio para una posterior comparativa con sus características teóricas. Y por otro mediante el desarrollo de un sistema que permita la captura, transmisión y publicación de datos de una aplicación específica empleando estos protocolos como método de intercomunicación.

Abstract

In a history moment defined by process of technological evolution growing, the concept of The Internet of Things arises in which an interaction between user and environment is achieved cause of the wireless communications technology. Giving to the users a better understanding of their environment.

In this field, the common ways to approach the concept of IoT are based on low-range technologies, traditional technologies such as Wi-Fi or the most recent option, new and free technologies, based on the concept of long-range and low consumption communications, in these kinds of technologies the battle for been chosen is taking place.

In these recent options, the LoRa technology and its standard LoRaWAN appeared, looking for developing IoT networks, with long distances transmissions and low consumption rates, they offer the user flexibility about the network design, with a well-formed community that supports its systems.

The project is made to get a deeper understanding of LoRa and LoRaWAN to verify its functionality and get to know their strengths and weaknesses.

In order to pursue it, the project focuses in two different ways to reach this goal. First, define the radio channel level to compare the results with the theory published. And in the other hand, developing a system that achieves to get, visualize and publish the data a specific application using these protocols for its achievement.

Índice

Índice de figuras	3
Índice de tablas	5
Índice de acrónimos y abreviaturas	6
1 Introducción	7
1.1 Motivación	7
1.2 Objetivos	8
1.3 Organización del trabajo	8
2 Estado del arte	9
2.1 LoRa	9
2.2 LoRaWAN	10
2.2.1 Topología de red	10
2.2.2 Esquema de protocolos	12
2.2.3 Clases de dispositivos	13
2.2.4 Dispositivos de clase A	13
2.3 Seguridad	19
2.3.1 Dispositivos de clase B	20
2.3.2 Dispositivos de clase C	21
2.4 Plataformas LoRaWAN	21
2.4.1 The Things Network	22
2.4.2 TheThings.IO	27
2.4.3 ResIOT	27
3 Diseño y evaluación	29
3.1 Equipamiento hardware	29
3.2 Red LoRaRAW	32
3.2.1 Configuración de los dispositivos	34

3.2.2	Caracterización del sistema.....	36
3.3	Sistema LoRa-MAC, nodo-Gateway.....	39
3.3.1	Configuración de los dispositivos.....	40
3.3.2	Caracterización del sistema.....	43
4	Sistema LoRaWAN	51
4.1	Arquitectura.....	51
4.2	Sección LoRa.....	52
4.2.1	Proceso de asociación	53
4.2.2	Recolección de información.....	53
4.2.3	Transmisión.....	54
4.3	Sección TTN	54
4.3.1	Gateway.....	54
4.3.2	The Things Network.....	56
4.4	Sección back-end/servidor.....	58
4.4.1	HTTPIntegration	58
4.4.2	Servicio Web.....	59
5	Conclusiones y líneas futuras.....	63
5.1	Conclusiones.....	63
5.1.1	Rendimiento y cobertura de redes LoRa	63
5.1.2	Funcionalidad y viabilidad en despliegues LoRa y LoRaWAN	64
5.2	Líneas futuras.....	65
6	Bibliografía	66

Índice de figuras

FIGURA 2.1 FUNCIONAMIENTO SPREADING FACTOR SOBRE LoRA	10
FIGURA 2.2 TOPOLOGÍA TRADICIONAL DE RED LoRA WAN	11
FIGURA 2.3 PILA DE PROTOCOLOS SOBRE LoRA WAN	12
FIGURA 2.4 VENTANA DE TRANSMISIÓN, CAPA PHY	14
FIGURA 2.5 MENSAJE UPLINK, DISPOSITIVO CLASE A	14
FIGURA 2.6 MENSAJE DOWNLINK, DISPOSITIVO CLASE B	14
FIGURA 2.7 TRADUCCIÓN DE NIVEL FÍSICO A NIVEL MAC DE FORMATO DE MENSAJES LoRA WAN	15
FIGURA 2.8 FORMATO CAPA MAC LoRA WAN	15
FIGURA 2.9 FHDR SUBDIVIDIDO CAPA MAC LoRA WAN	15
FIGURA 2.10 FORMATO DIRECCIONES EN LoRA WAN	17
FIGURA 2.11 FORMATO JOIN-ACCEPT LoRA WAN	18
FIGURA 2.12 PROCESO DE INSERCIÓN DE SEGURIDAD Y AUTENTICACIÓN	19
FIGURA 2.13 SINCRONIZACIÓN DISPOSITIVOS CLASE B	20
FIGURA 2.14 ESCUCHA EN DISPOSITIVOS CLASE C	21
FIGURA 2.15 DISPERSIÓN GLOBAL DE TTN	23
FIGURA 2.16 GW SUBSCRITO A TTN	24
FIGURA 2.17 INTERFAZ DE DECODIFICACIÓN EN TTN	25
FIGURA 2.18 ARQUITECTURA OFERTADA POR THE THINGS.IO	27
FIGURA 2.19 ARQUITECTURA DE RED SOBRE REST.IO	27
FIGURA 3.1 HARDWARE - PLACA LoPY	30
FIGURA 3.2 INTERFAZ ATOM Y CONSOLA REPL	31
FIGURA 3.3 PINOUT LoPY	31
FIGURA 3.4 EXPANSION BOARD 3.0	32
FIGURA 3.5 PYTRACK	32
FIGURA 3.6 TRANSMISIÓN LoRARAW	33
FIGURA 3.7 PROCESO TRANSMISIÓN-SISTEMA LoRARAW	34
FIGURA 3.8 RECEPCIÓN LoRARAW	36
FIGURA 3.9 PUNTOS DE MEDICIÓN EN CARACTERIZACIÓN SISTEMA LoRARAW	37
FIGURA 3.10 CARACTERIZACIÓN SISTEMA LoRARAW	38
FIGURA 3.11 ESQUEMA SISTEMA LoRAMAC	40
FIGURA 3.12 TRANSMISIÓN SISTEMA LoRAMAC	41
FIGURA 3.13 RECEPCIÓN SISTEMA LoRA-MAC	42
FIGURA 3.14 FORMATO DATOS LoRA-MAC	42
FIGURA 3.15 FORMATO ACK LoRA-MAC	42
FIGURA 3.16 CARACTERIZACIÓN LoRA-MAC EN PRIMERA MEDICIÓN	44
FIGURA 3.17 DISTANCIA/SNR. CAMPAÑA DE MEDICIÓN SON SF VARIABLE, SF=7	45
FIGURA 3.18 CAMPAÑA DE MEDICIÓN CON SF VARIABLE, SF=7	45
FIGURA 3.19 CAMPAÑA DE MEDICIÓN SON SF VARIABLE, SF=9	47
FIGURA 3.20 CAMPAÑA DE MEDICIÓN SON SF VARIABLE, SF=12	49
FIGURA 4.1 ESQUEMA SISTEMA LoRA WAN	52
FIGURA 4.2 ESQUEMA DE TRANSMISIÓN LoRA WAN	53
FIGURA 4.3 FORMATO PAQUETE TRANSMITIDO LoRA WAN	54
FIGURA 4.4 ACTIVACIÓN DEL NANOGATEWAY	56
FIGURA 4.5 REGISTRO NANOGATEWAY	57

FIGURA 4.6 REGISTRO NANOGATEWAY	57
FIGURA 4.7 CONFIGURACIÓN HTTP INTEGRATION.....	59
FIGURA 4.8 ESQUEMA DE FUNCIONAMIENTO DEL BACK-END DE RED LORAWAN	59
FIGURA 4.9 TABLA MYSQL DE APLICACIÓN REST	60
FIGURA 4.10 INTERFAZ USUARIO CON SERVICIO WEB(/).....	61
FIGURA 4.11 INTERFAZ USUARIO CON DATOS(/NAMES).	61
FIGURA 4.12 GRÁFICAS AUTORREGULADAS EN SERVICIO WEB	61

Índice de tablas

TABLA 1 SERVICIOS EXTERNOS A TTN.....	26
TABLA 2 ESPECIFICACIONES LOPY	30
TABLA 3 DESPLIEGUE CARACTERIZACIÓN SISTEMA LORARAW	37
TABLA 4 -MEDIA RESULTADOS CARACTERIZACIÓN SISTEMA LORARAW.....	38
TABLA 5 RESULTADOS 1ª CAMPAÑA DE MEDIDA LORA-MAC.....	43
TABLA 6 CAMPAÑA DE MEDICIÓN SON SF VARIABLE, SF=7.....	45
TABLA 7 CAMPAÑA DE MEDICIÓN SON SF VARIABLE, SF=9.....	46
TABLA 8 CAMPAÑA DE MEDICIÓN SON SF VARIABLE, SF=12.....	48
TABLA 9 PARÁMETROS DE CONFIGURACIÓN NANOGATEWAY.....	55

Índice de acrónimos

IoT	Internet of Things
NB-IoT	Narrowband-IoT
LTE Cat 1	Long Term Evolution Cat 1
WiFi	Wireless Ethernet Compatibility Alliance
LoRa	Long Range
SF	Spreading Factor
LoRaWAN	Long Range Wide Area Network
GW	Gateway
NS	Network Server
AS	Application Server
PHY	Física
OTAA	Over-The-Air-Activation
ABP	Activation-by-Personalization
MAC	Media Access Control
ADR	Adaptative Data Rate
TTN	The Things Network
DR	Data Rate
NwkSKey	Network Server Key
AppSKey	Application Server Key
ADC	Analog-to-Digital Converter
LDR	Light-dependent Resistor
NTP	Network Time Protocol
API	Application Programming Interface

1 Introducción

La tecnología evoluciona en función de las necesidades que las personas requerimos de ella y a medida que el tiempo pasa, nuestros requerimientos van aumentando y se le atribuyen nuevas finalidades a esta. En los últimos años las tecnologías de comunicación inalámbrica han ido cambiando mucho, con numerosas tecnologías quedando prácticamente obsoletas. La innovación cada más rápida y con resultados más eficientes es la principal causa de este proceso.

Acompañando a esta evolución, en cierto punto surge el concepto de Internet de las Cosas (IoT, Internet of Things)[1], el cual tiene como propósito una mayor interconexión del usuario con su entorno, estableciendo redes de comunicación que habiliten la interacción con el contexto cercano del usuario, redundando en la captura de datos de los dispositivos a su alrededor.

Esta perspectiva gana adeptos y múltiples tecnologías son desplegadas. Entre ellas, tecnologías en manos de operadoras son ofertadas como “Narrowband-Internet of Things” (NB-IoT, Narrowband-IoT), o “Long Term Evolution Cat 1” (LTE Cat 1, Long Term Evolution Cat 1), las cuales han ido ganando fuerza.

También se puede mencionar y se emplean tecnologías que no habían sido diseñadas inicialmente para este nuevo paradigma de red como WiFi o las antiguas tecnologías de telefonía móvil, que tratan de adaptarse en este concepto.

En contraposición a ellas, surgen otras que además de operar en bandas de frecuencias que no requieren de licencia, se adaptan mejor a los requisitos de la IoT. Esto supone inicialmente un coste menor y mejor operativa del despliegue, al tiempo que facilita la expansión de la red mediante despliegues abiertos y colaborativos.

Tecnologías como Sigfox [5] o LongRange (LoRa, Long Range), buscan otorgar la capacidad a todo el mundo de poder crear una red IoT, siendo este conjunto de tecnologías definidas por crearse en torno a dos características, bajo consumo y amplia cobertura.

1.1 Motivación

Las redes IoT, como se ha descrito, son un paradigma de red en gran auge actualmente. LoRa junto LoRaWAN se colocan en una posición de disponibilidad para cualquier usuario que quiera desarrollar estas redes.

Tenemos así una tecnología que podría considerarse nueva en un entorno tecnológico cada vez más usado. Esta tecnología oferta una cobertura amplia y un consumo muy bajo. De corroborarse que los despliegues cumplan y se demuestren su completa funcionalidad, estaríamos ante una opción más que adecuada para trabajar sobre el concepto IoT, y que la pondría como opción preferente entre las que actualmente compiten por liderar los despliegues IoT.

1.2 Objetivos

En consecuencia, a lo anteriormente descrito, para poder confirmar la viabilidad de LoRa se busca en este proyecto:

- Aplicar la comunicación bajo el protocolo LoRa entre dispositivos.
- Diseño e implementación de un sistema bajo la tecnología LoRa.
- Verificación de las virtudes de la tecnología LoRa a través de su caracterización.
- Diseño y despliegue de un entorno IoT que haga uso de LoRa y LoRaWAN, que habilite la captura, almacenamiento y procesamiento de datos y la puesta a disposición de estos al usuario final.

1.3 Organización del trabajo

El proyecto aquí descrito se subdivide en 5 partes, véase;

- En primer lugar ‘Introducción’ en la que se esboza el contenido total del proyecto a modo de guía para un mejor entendimiento de este.
- Una segunda parte, ‘Estado del arte’, que contendrá el marco teórico que engloba al proyecto, tanto los protocolos involucrados, así como una descripción de la plataforma de la que se hará uso en este.
- En tercer lugar, ‘Diseño y evaluación’ donde se hará una evaluación la tecnología LoRa a través de la implementación de despliegues sencillos.
- Para completar el proyecto, se trabajará y desarrollará un sistema completo sobre la tecnología LoRaWAN. Este proceso se encuentra descrito en el capítulo ‘Sistema LoRaWAN’.
- ‘Conclusiones y líneas futuras’ hace uso de todo el contenido que se ha postulado para dar un veredicto a modo de respuesta a la motivación de este proyecto, y se añade expectativas y posibilidades a futuro que podrían llevarse en los múltiples desarrollos descritos.

2 Estado del arte

En este capítulo se tratará de desarrollar todos los conceptos teóricos necesarios para la comprensión del proyecto. Por ello se detallará en primer lugar los protocolos LoRa y LoRaWAN.

También se expondrán las plataformas basadas en esta tecnología, dando ejemplos sobre estas, y detallando en mayor profundidad The Things Network, plataforma con la que se trabajará en este proyecto.

2.1 LoRa

LoRa[2] es lo que conocemos por una serie de características bien definidas agrupadas que definen una tecnología de capa física sobre la que se asientan protocolos de niveles superiores para la transmisión de radiocomunicaciones inalámbricas enfocadas a un bajo consumo.

Se constata como una tecnología adecuada para la creación de redes de bajo consumo otorgando a estas un amplio radio de funcionamiento, alcances entre 15-20Km para conexiones punto a punto con línea de visión directa. No obstante, hay que hacer hincapié en el decrecimiento de este valor en áreas que no cumplan esta condición de visión entre dispositivos, así como una pérdida aún mayor de cobertura en entornos urbanos.

Las transmisiones LoRa[4] operan en las bandas ISM gratuitas determinada en función a la zona geográfica en la que se opera; véase 868MHz en Europa, 433 en Asia y 915Mhz en América.

En contraparte posee unas tasas binarias de funcionamiento bajas, delimitado por poder enviar hasta unos 255 bytes por transmisión llegando a adquirir desde 250 bps hasta 11kbbps.

La variación en tasas binarias, así como en alcance obtenido son parámetros a nivel físico modulados en función al Spreading Factor [3], siendo la característica clave que define una transmisión.

El Spreading Factor (SF) es aplicado en cada transmisión, siendo configurable por el usuario. Aporta la opción de realizar envíos lentos y claramente para aumentar la distancia de cobertura que se posee, o bien beneficiar las transmisiones cercanas. Por defecto el SF tiene el valor 7 y puede oscilar hasta un máximo de 12.

Tal como se describe en la Figura 2.1, la variación del SF genera cambios en la tasa binaria de la transmisión, el tamaño máximo del contenido de un paquete y la sensibilidad mínima a la que un receptor puede captar un mensaje. Podemos observar que en función al aumento del SF en una transmisión, se produce una disminución del tamaño máximo de datos, y de la tasa binaria

(oscilando de 222 bytes de payload a 11kbps, hasta 51 bytes a 250 bps¹). Mientras que la sensibilidad a la que es capaz de captar mejora en función al aumento del SF. El aumento de esta permite la recepción de señales con una peor calidad oscilando entre -120dBm a -136dBm.

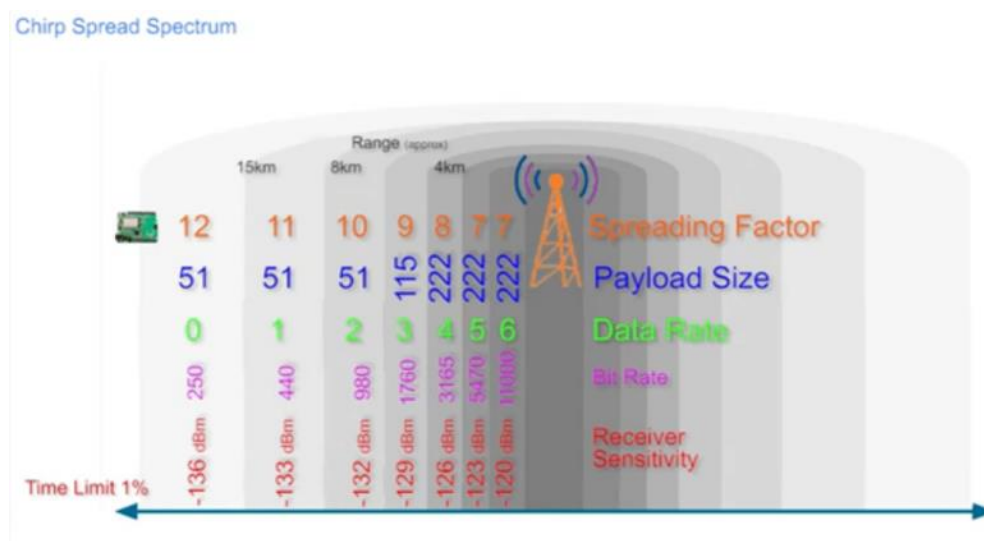


Figura 2.1 Funcionamiento Spreading Factor sobre LoRa

Sobre esta capa física se trabaja pues en busca de un protocolo que dadas las características tendrá una gran utilidad en la implementación de redes IoT definidas por la necesidad de un bajo consumo y no requerir de unas tasas binarias elevadas.

También ha de mencionarse que el estándar LoRa no está abierto por ello no se poseen todos los detalles técnicos sobre el modo de funcionamiento. Por el contrario, LoRaWAN es un estándar abierto y disponible al público.

2.2 LoRaWAN

LoRaWAN (LoRaWAN, Long Range Wide Area Network) es el anexo que surge a la tecnología de capa física LoRa. Es un protocolo de red que establece las capas superiores a LoRa, el cual se encarga de la mayoría de las funcionalidades en las comunicaciones como la seguridad, el control de acceso al medio, establecimiento de formato de tramas, etc.[6]

2.2.1 Topología de red

Una red LoRaWAN permite comunicaciones bidireccionales entre todos los componentes que la conformen, así como la posibilidad de confirmación en los mensajes. En la Figura 2.2 se muestra la topología de red más habitual.

¹ Payload: carga de datos útil en un paquete.



Figura 2.2 Topología tradicional de red LoRaWAN

Los elementos que conforman la red LoRaWAN son:

- **End-Nodes Devices;** nodos de comunicación con el Gateway de la red.
- **Gateways (GW);** puente entre la red LoRaWAN y una red externa encargándose de la recolección de mensajes de los dispositivos end-node y reenvío a la red externa o viceversa.
- **Network Server (NS);** parte final del sistema, gestiona los mensajes de control y autentica los mensajes recibidos. En caso de la existencia de un servidor de aplicación gestiona la transmisión de los datos a este.
- **Application Server (AS),** servidor opcional sobre el que se gestionará el nivel de aplicación.

Si se analiza la arquitectura de red se pueden observar dos zonas bien diferenciadas. Por un lado una zona LoRa conformada por las comunicaciones entre el GW y los diversos end-devices de la red, y por otro, una zona que se puede considerar de comunicaciones IP o Internet, donde el GW hace de puente y transmite los datos empaquetados hacia el NS y posteriormente al AS.

Además, se ha de añadir que el uso de otras tecnologías como Bluetooth o WiFi puede ser adherida en una etapa previa a los End-Node Devices para la obtención de los datos que serán transportados por el sistema LoRaWAN.

Si bien los nodos transmiten desde cualquier canal que tengan disponible a la tasa binaria que puedan, para el correcto funcionamiento de la red LoRaWAN los nodos deben cumplir los siguientes requisitos:

- Se realice un cambio pseudoaleatorio de canal en cada transmisión para generar un sistema más robusto.

- Se respete el duty_cycle ²máximo establecido por las regulaciones de la región, que para el caso de Europa se corresponde al 1%[8].
- Se respete el tiempo máximo de transmisión establecido según las regulaciones de la región.

2.2.2 Esquema de protocolos

LoRaWAN junto con LoRa conforma una pila de protocolos la cual da vida a un modelo de comunicaciones. En la Figura 2.3 se resume la pila de protocolos definida para cada uno de los niveles.

A nivel físico, se basa en el empleo de la tecnología LoRa que opera en la banda ISM ya mencionada en la sección 2.1.

Una vez se tiene la capa física operativa, LoRaWAN conforma el nivel inmediatamente superior, el cual se encarga del acceso al sistema, así como gestión de los dispositivos que conforman la red, y sus mensajes, y la seguridad del sistema a nivel MAC. Dentro del nivel MAC, LoRaWAN especifica tres clases diferentes de dispositivos, clase A, B y C, que afectan a las cualidades vinculadas a la transmisión y recepción de mensajes, incluyendo reconocimientos de mensajes, ocupación de canal, etc.

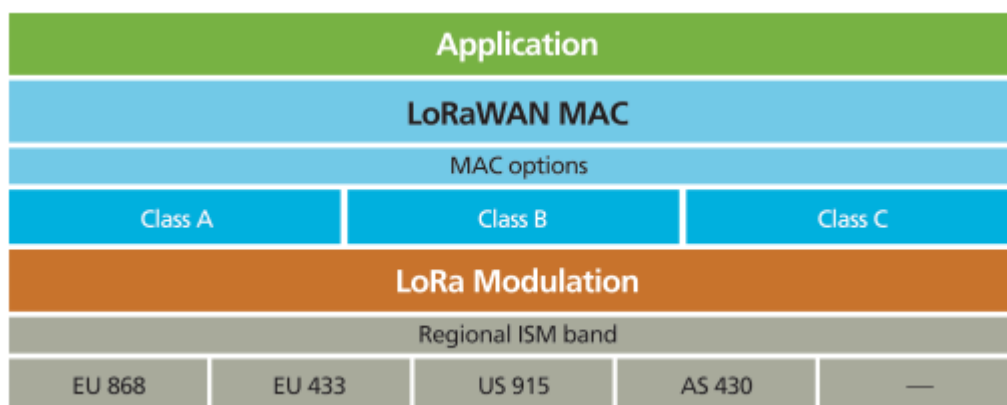


Figura 2.3 Pila de protocolos sobre LoRaWAN

Por último, se tiene una capa superior de aplicación que se gestiona cada end-device por sí mismo, ya que es dependiente de la tarea que vaya a realizar.

Los GW implementan una pila dual que incluye, además de la relativa a LoRa, los niveles de la pila TCP/IP tradicional. Esto es debido a que los GWs han de hacer de puente hacia el NS.

² Duty_cycle: es la relación entre el estado activo que tiene una señal, frente al periodo total de esta.

2.2.3 Clases de dispositivos

Tal como se ha descrito, LoRaWAN define tres clases de dispositivos:

- **Dispositivo de clase A**
Permiten la comunicación bidireccional respecto a cada transmisión. Una vez se hace una transmisión en el enlace ascendente (uplink) desde el end-device, se esperan dos ventanas cortas de recepción en el descendente (downlink).
Básicamente el modelo de acceso al medio sigue un sistema similar a Aloha[18] donde cada comunicación se hace en función a las necesidades propias con un tiempo de espera aleatorio.
Esta clase es la que posee un consumo de batería menor, por ello son más susceptibles de ser dispositivos que solo requieran del envío de mensajes en sentido ascendente sin mucha participación en la comunicación por parte del NS.
- **Dispositivo de clase B**
Son el paso inmediatamente superior a la clase A, e implementa el uso de una agenda predefinida de tiempo de escucha. Se recurre al intercambio de beacons³ con el GW para la sincronización de las diversas agendas, de forma que el servidor siempre sabe cuándo los end-devices están escuchando el canal.
- **Dispositivo de clase C**
Son la clase de nivel superior. Frente a la aleatoriedad o predeterminación de periodos de escucha, se opta por una escucha continuada del medio, la cual solo es cerrada para hacer un periodo de transmisión.
Se ofrece así la menos latencia entre servidor y nodos a cambio de un mayor consumo energético. Por ello es recomendable usarlo solo en dispositivos que dispongan de un modo de mantener su batería cargada.

El funcionamiento del sistema se ve mayormente fomentado por los tipos de dispositivos que se implementen, por ello se procede a describir el protocolo en función a los tipos de dispositivo.

2.2.4 Dispositivos de clase A

Todo dispositivo que pasa a conformar parte de la red LoRaWAN parte siendo un dispositivo de clase A. Por tanto, todos los procedimientos a continuación descritos son comunes a todos los

³ Beacon: paquete transmitido con información referente a la sincronización entre las agendas de diversos dispositivos.

dispositivos, al menos inicialmente. Una vez unidos a la red, los dispositivos tienen la capacidad de negociar a través de su Gateway para cambiar la clase de dispositivo con la que es considerado en caso de querer actuar como un dispositivo de clase B o C.

Los dispositivos de clase A serán los utilizados en este proyecto, por ello se entra en mayor detalle en su funcionamiento, y siendo además la clase inicial de todo dispositivo.

Capa PHY

Tal como se ha introducido, y según se representa en la Figura 2.4, tras cada transmisión uplink se hace una espera de dos períodos cortos de escucha

2.2.4.1

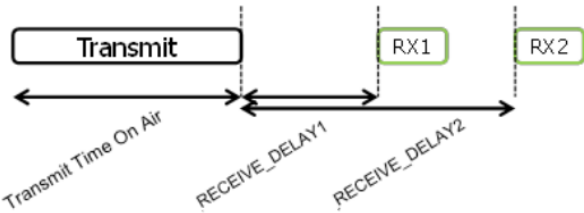


Figura 2.4 Ventana de transmisión, capa PHY

Los mensajes uplink se envían desde los end-devices a cualquier Gateway disponible. Estos mensajes incluyen además del conjunto de datos de usuario/aplicación (PHYPayload), un preámbulo, una cabecera propia de LoRa (PHDR) y su autenticador (PHDR_CRC). El mensaje completo se encuentra protegido por un CRC. En la Figura 2.5 se representa un mensaje de uplink para los dispositivos de clase A.

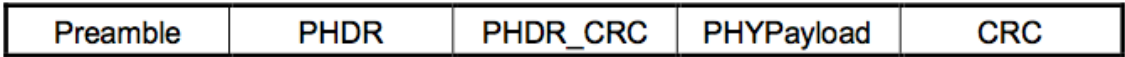


Figura 2.5 Mensaje uplink, dispositivo clase A

Por su parte, los mensajes downlink; representado en la Figura 2.6, son enviados desde el servidor a través de un único GW en modo punto a punto hacia el end-device. El formato es similar al mensaje uplink, pero sin protección CRC para los datos.

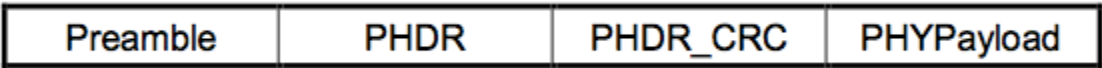


Figura 2.6 Mensaje downlink, dispositivo clase B

El período de escucha para la recepción de mensajes ha de ser, como mínimo, lo suficientemente largo como para que permita leer la totalidad del preámbulo de un mensaje de capa PHY. Las características de ambas ventanas de recepción son diferentes: mientras que el primer periodo de escucha usa una frecuencia y tasa binaria para la recepción igual a aquella utilizada por el mensaje uplink, el segundo periodo es configurable para poder usar otros.

No obstante, es importante señalar que una vez se ha recibido y verificado un mensaje correcto en el end-device en el primer período de escucha, el segundo período es descartado.

Capa MAC

Los mensajes de formato PHY se conforman a partir del payload obtenido de la capa MAC, tal como se describe en la Figura 2.7. Los mensajes a este nivel incluyen una cabecera (MHDR), un campo de carga útiles que encapsulará los datos, y un código de integridad de los datos (MIC).

2.2.4.2

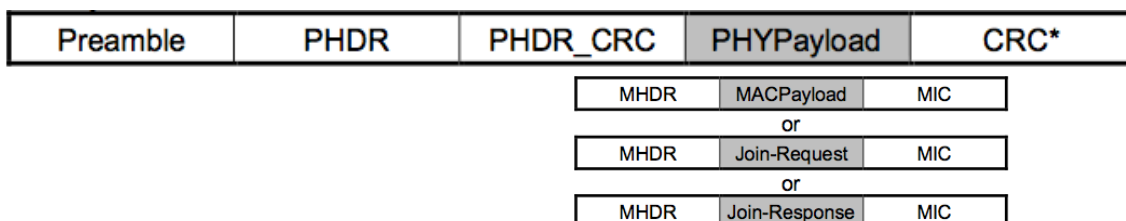


Figura 2.7 Traducción de nivel físico a nivel MAC de formato de mensajes LoRaWAN

A su vez el payload de capa MAC está conformado por una cabecera de trama (FHDR), y dos campos opcionales, uno que define el puerto (FPort) y el propio campo de carga útil de la trama (FRMPayload). En la Figura 2.8 se observa dicha trama.



Figura 2.8 Formato capa MAC LoRaWAN

El campo cabecera de trama FHDR (Figura 2.9) se compone de:

- DevADDR: la dirección del dispositivo transmisor
- FCtrl: campo de control que incluye información que permite gestionar los siguientes modos de operación:
 - Requerimiento y transmisión de ACK en caso de comunicación con confirmación
 - Activación del Adaptive Data Rate (ADR), el cual permite al Network Server ser el que controle la tasa binaria de funcionamiento del End-Device
 - Identificar si la trama transporta claves.
 - Identificar si hay mensajes pendientes por parte del servidor.
- FCnt: campo que actúa de contador de trama. Incluye un contador para los mensajes en sentido ascendente (FCntUp) y otro para el sentido descendente (FCntDown)
- FOpts: campo incluye opciones y/o comandos de nivel MAC.

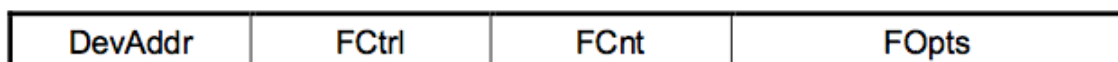


Figura 2.9 FHDR subdivido capa MAC LoRaWAN

Por su parte, el campo FPort, siempre debe aparecer en caso de haber payload en el mensaje a nivel MAC. Los valores son específicos de cada aplicación, a excepción del valor 0 que indica que los datos en FRMPayload solo contienen comandos MAC.

Por último, FRMPayload contiene el payload a nivel MAC, que se encripta antes del cálculo del MIC con una clave en función de si el mensaje está destinado al NS o el AS

Se diferencian así 8 tipos de datos o tramas MAC:

- Join request/Join Accept.
- Data Uplink/Downlink Confirmed; requiere de ACK del transmisor.
- Data Uplink/Downlink Unconfirmed; no requiere de confirmación del receptor.
- RFU; para el intercambio de claves.
- Un último tipo propietario.

A continuación, se describen los diferentes comandos, así como el procedimiento de activación y enganche a la red LoRaWAN de un end-device.

2.2.4.2.1 Comandos MAC

A nivel de gestión de red, existen un conjunto de comandos MAC que se intercambian entre el NS y los end-devices y que no son visibles por el AS o la propia aplicación. Estos comandos vienen indicados por un valor en el campo FPort a 0 junto con los diversos datos que se deseen mandar. Entre ellos caben destacar:

- Link Check Command: comando utilizado y transmitido por los end-devices que sirve para validar su conexión con el NS. Una vez recibido por el NS, este le devuelve el mismo comando con la sensibilidad en dBm con la que ha recibido el mensaje.
- Link ADR Command: implementado por el NS, lo envía a un end-device para implementar ADR sobre él, es decir, permite al NS determinar la máxima potencia de transmisión que usará el end-device, así como que canales debe usar para las transmisiones uplink y la tasa binaria de trabajo.
- Duty cycle de un end-device: permite al NS determinar un nivel máximo de duty cycle al que puede trabajar el end-device con el que se comunique. En caso de enviarle el campo vacío se asume que no hay limitación.
- Receive Windows Parameters: permite al NS gestionar la segunda ventana de escucha que poseen los end-devices, pudiendo configurar la frecuencia y la tasa binaria a la que esta funciona.
- Creation Modification of Channel: sirve para la creación de nuevos canales de transmisión, así como para la modificación de los parámetros de los canales existentes.

- Setting Delay between TX and RX: permite al NS enviar un comando a los end-devices para poder cambiar el tiempo entre el periodo de transmisión y el comienzo de periodo de recepción.

El uso de todos los comandos permite hacer flexible a las diversas necesidades del sistema las características de los canales y end-devices que lo componen.

Activación de un end-device

Para poder establecer una comunicación desde un end-device, ya sea con el GW o con el NS, se requiere de un proceso previo de activación, el cual puede llegar a conseguirse por 2.2.4.3 Over-The-Air-Activation (OTAA) o Activation-By-Personalization (ABP).

La exitosa finalización de sendos procesos de activación resulta en el establecimiento en los end-devices de una serie de parámetros:

- Dirección del end-device (DevAddr): identificador de 32 bits del end-device dentro de la red. Está compuesto por 5 bits identificadores de red (NwkID) y 25 bits con la dirección única de cada dispositivo (NwkAddr). Puede observarse en la Figura 2.10 como se conforma este campo.

Bit#	[31..25]	[24..0]
DevAddr bits	NwkID	NwkAddr

Figura 2.10 Formato direcciones en LoRaWAN

- Application Identifier (AppEUI): identificador global de la aplicación a la que un end-device pertenece o para la que opera (i.e operador).
- Application Session Key (AppSKey): clave de sesión compartida entre el AS y el end-device, la cual permite mantener los datos encriptados que intercambian entre ambos confidenciales para el resto de los componentes de la red. Se compone de 128 bits y se preestablece en antes de la activación.
- Network Session Key (NwSKey): clave de sesión compartida entre el end-device y el NS para el cifrado y autenticación de datos entre ambos. Con una longitud de 128 bits se preestablece antes de la activación en el dispositivo.

2.2.4.3.1 Over The Air Activation

Para aplicar el proceso de activación OTAA, los end-devices deben completar un proceso de unión/registro antes de proceder con las comunicaciones. Cada vez que se pierda información de la sesión se ha de reinicializar dicho proceso.

OTAA requiere que el end-device haya sido configurado con el identificador global y único del dispositivo (DevEUI), el identificador de la aplicación (AppEUI) y una clave maestra para la

aplicación (AppKey). Los procedimientos criptográficos para este proceso se realizan empleando AES-128.

El proceso, iniciado siempre por el end-device, se compone del intercambio de dos mensajes, Join-Request transmitido por parte del end-device y Join-Accept por parte del NS:

- Join-Request

Enviado por el end-device hacia el NS, la trama MAC está compuesta de la dirección del dispositivo (DevEUI), un parámetro de acceso a aplicación (AppKey), y un valor aleatorio (DevNonce). El contenido se cifra con la AppKey y a su vez se genera un MIC para su autenticación.

- Join-Accept

Enviado desde el NS hacia el end-device como resultado del proceso de asociación, incluye la siguiente información (Figura 2.11):

- AppNonce, valor aleatorio de aplicación
- DevAddr, la dirección del dispositivo
- NetID, ID de la red en que se trabaja
- CFList, lista de canales con frecuencias adicionales que puede usarse
- RxDelay, el retardo que desea entre transmisión y escucha del end-device.

Size (bytes)	3	3	4	1	1	(16) Optional
Join Accept	AppNonce	NetID	DevAddr	DLSettings	RxDelay	CFList

Figura 2.11 Formato Join-accept LoRaWAN

El valor AppNonce se utilizará para la generación de una NwkSKey y AppSKey únicas, consiguiendo así que no requieran de estar preestablecidas en el componente, sino que han sido resultado del proceso de asociación con el NS.

2.2.4.3.2 Activation By Personalization

Se utiliza para hacer una activación sin tener que negociar todos los parámetros requeridos con el NS y derivarlos de otros valores. Se posee una DevAddr, AppSKey y NwkSKey almacenadas antes de la activación en el dispositivo.

El NS solo verificará que estas son correctas permitiendo la activación del end-device con un proceso de JoinRequest y JoinAccept similar a OTAA.

2.2.4.3.3 Seguridad

Común a cualquier modo de activación o clase de dispositivo, la implementación del protocolo LoRaWAN conlleva la aplicación de procesos de seguridad sobre las transmisiones que se realicen.

Este proceso se lleva a cabo a dos niveles diferentes en cada dispositivo, implementando un cifrado del payload junto a una verificación de los datos transmitidos. Estos niveles son:

- **Nivel de aplicación:**

El payload encapsulado a nivel MAC que será redirigido hasta el AS se cifra usando AES128 con la clave de 128 bits AppSKey generada, lo cual mantiene los datos no visibles exceptuando al propio end-device y al AS que poseen la clave para poder descryptarlo.

- **Nivel de Red:**

Se aplica un cifrado y autenticación de los datos. Se recurre a la NwkSKey para el cifrado de los datos a nivel MAC intercambiados entre el end-device y el NS, añadiendo una cabecera final en función de estos MIC para verificar que no se haya modificado el contenido original.

En la Figura 2.12 se incluye un diagrama de la generación de un mensaje y cómo se integran los mecanismos de seguridad.

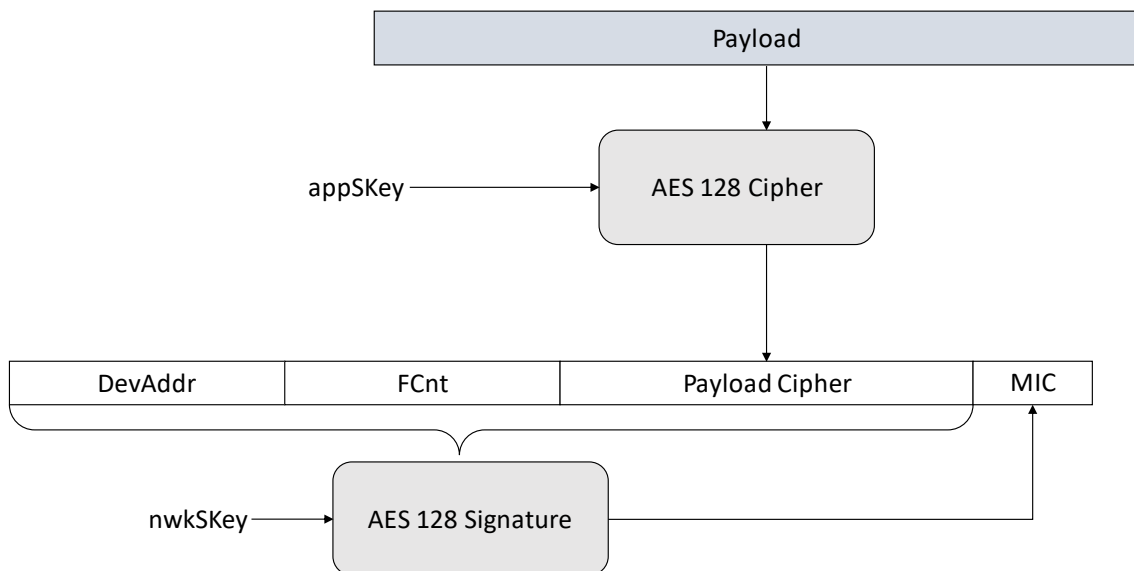


Figura 2.12 Proceso de inserción de seguridad y autenticación

2.2.5 Dispositivos de clase B

Frente al empleo de un modelo de acceso al medio similar a ALOHA de los dispositivos de clase A, los dispositivos de clase B se distinguen por tener tiempos prefijados de escucha, basándose en la escucha de beacons enviados por el GW que habilitan la sincronización entre todos los end-devices que conforman la red, generando cada dispositivo su propia agenda.

Por tanto, el modo de operación del canal descendente requiere de una sincronización. El resto del funcionamiento de la red tiene un comportamiento similar a un dispositivo de clase A.

Hasta el momento no hay ningún servicio disponible que soporte esta clase de dispositivo, aunque esté definido en el estándar.

Sincronización entre dispositivos:

2.2.5.1 Todo GW de la red ha de tener la capacidad de enviar un mensaje broadcast a los end-devices con los beacons generados por el NS.

La activación del nodo y la asociación a la red se implementa del mismo modo que en un dispositivo de clase A, hasta que el nodo comunique al NS que quiere cambiar su clase a B, en cuyo caso se emplearán los beacons para sincronizarse con la red.

Estos beacons permiten a los end-devices saber cuándo pueden abrir su periodo de escucha para el inicio de una recepción de datos en el canal descendente, referido como “ping slots”, referenciando a su vez a estas comunicaciones que el NS envía como “ping”.

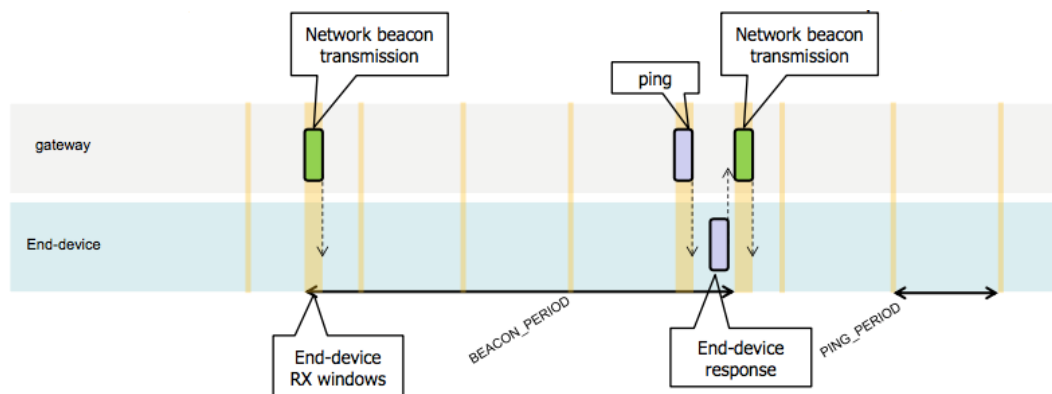


Figura 2.13 Sincronización dispositivos clase B

Conseguir este funcionamiento requiere que el NS disponga de la periodicidad, la tasa binaria y el canal en el que se implementen los ping-slots.

Una vez sincronizados, los NS poseerán la capacidad de variar cualquiera de las propiedades del ping-slot de un end-device. A su vez el end-device periódicamente informará al NS de su localización en la red para mantener un esquema de la topología actualizado. Este proceso puede verse representado en las transmisiones de la Figura 2.13.

Hay que señalar que en caso de que un end-device no consiga la recepción de un beacon tras una ventana de tiempo determinada, perderá su condición de clase B actual y retomará el funcionamiento como un dispositivo de clase A.

2.2.6 Dispositivos de clase C

Un end-device operando como clase C se caracteriza por mantenerse siempre en escucha en la red, por lo que excluye que pueda implementar a su vez un perfil de clase B.

El esquema del perfil clase A exponía la existencia de dos ventanas de escucha, siendo la segunda una ventana configurable por el NS. Los dispositivos que implementan la clase C se mantienen todo el tiempo posible escuchando en esta segunda ventana, exceptuando los momentos en que han decididos transmitir o escuchar por la primera ventana. Para ello, abren una corta ventana temporal entre la finalización de la transmisión de datos a través del canal ascendente y el inicio de la primera ventana de escucha, en la que reciben parámetros de las características de la segunda ventana. Tan pronto finaliza el primer periodo de escucha, aplican los parámetros de recepción para operar en la segunda ventana, en la que se mantienen hasta que tienen que volver a transmitir datos en el enlace ascendente. En la Figura 2.14 se observa la operativa de los dispositivos clase C.

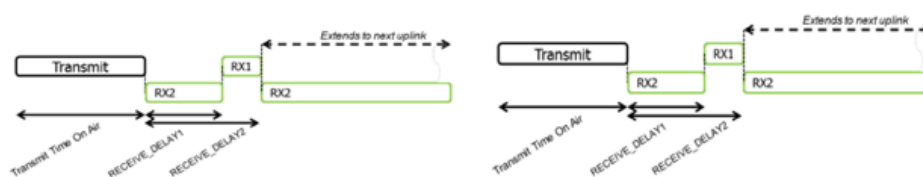


Figura 2.14 Escucha en dispositivos clase c

Al igual que en el caso del perfil B, la unión a la red y activación se hace como dispositivo de clase A, tras lo cual se posibilita notificar al NS la elección de hacer uso de funcionamiento de un dispositivo de clase C.

Aunque la implementación de dispositivos de clase C puede llevarse a cabo, no sería recomendable, pues uno de los propósitos principales de los despliegues basados en este estándar es poder otorgar dispositivos con un bajo consumo, y el mantenimiento constante entre el estado de transmisión y recepción de los dispositivos clase C los hace menos eficientes energéticamente. Por ello tendrán generalmente un requerimiento de tener una alimentación fija a ellos, con mucha más capacidad respecto a los dispositivos A o B.

2.3 Plataformas LoRaWAN

Uno de los puntos fuertes de la tecnología LoRa y LoRaWAN definida anteriormente es que ante su crecimiento ha surgido una serie de plataformas que buscan dar apoyo a esta metodología de trabajo.

Estas plataformas nacen con la idea de un trabajo colaborativo para la creación de una gran comunidad donde se interconecten todos los sistemas que la constituyen. Estas plataformas pasan a conformar la base de sustento del sistema LoRaWAN sobre el que se implemente, ya que la plataforma pasará a dar lugar a el NS y AS de la red. Facilitando el desarrollo de sistemas basados en esta tecnología en gran medida si se recurre a estas alternativas.

Una de las grandes plataformas que constituyen esta capa de soporte a LoRaWAN es The Things Network, plataforma bajo la cual se ampara este proyecto, aunque existen otras plataformas que operan bajo la tecnología LoRaWAN como TheThings.io o Rest.io entre otras.

2.3.1 The Things Network

Ante el surgir de la tecnología LoRa y la implementación de LoRaWAN en múltiples redes, surge la iniciativa The Things Network (TTN) [7] como una red open-source, accesible y abierta a todo el mundo, que trabaja sobre tecnología LoRa.

TTN es creado en el año 2015 como un método para facilitar el diseño y creación de redes LoRaWAN aportando a sus usuarios todo el back-end de la red, es decir, una implementación de las funcionalidades del NS y AS. A su vez TTN oferta diversas herramientas y métodos aplicables sobre su web, que permiten generar servicios a nivel de aplicación de manera sencilla para todo aquel que quiera crear su propia red LoRaWAN.

Por su carácter de desarrollarse como una open-source y el hecho de haber optado por una tecnología la cual está en pleno proceso de crecimiento como es LoRaWAN, TTN ha desarrollado una amplia comunidad de usuarios a lo largo del mundo, tal como se observa en el mapa incluido en la Figura 2.15. TTN posibilita la idea de comunidades conformadas por usuarios que pertenezcan a una zona geográfica relativamente semejante, habilitando que puedan juntarse como grupo de usuarios que comparten los datos generados entre sí. Actualmente se ha llegado a superar las 400 comunidades. Cabe destacar el múltiple número de comunidades que se han creado en este reciente tiempo en España para ver el crecimiento que esta tecnología está sufriendo. Actualmente se encuentran disponibles comunidades en Barcelona, Madrid, Tarragona, Castellón, Bilbao o Asturias entre otras [7].



Figura 2.15 Dispersión global de TTN

Funcionamiento

2.3.1.1 The Things Network es hoy compatible para el uso de LoRaWAN. Se requiere de un registro previo para poder usarla, lo cual habilita la definición de aplicaciones, y la activación de GW y end-devices que formarán parte de la red/comunidad.

Poseer un módulo LoRaWAN certificado, como es el caso de la placa LoPy de Pycom [15] así como los microchips RN [11], es uno de los requisitos obligatorios para poder conectarse a la plataforma.

Para una correcta adhesión, se requiere:

- Implementar un método de Joining ABP u OOTA en los end-devices similar al descrito por el protocolo estandarizado y descrito en la sección 2.2.4.3.
- Almacenar en TTN las claves necesarias de los diversos dispositivos, según sea el método de activación.
- Asignar una descripción sobre los dispositivos que van a implementarse y asignarles una dirección en la red DevEUI.

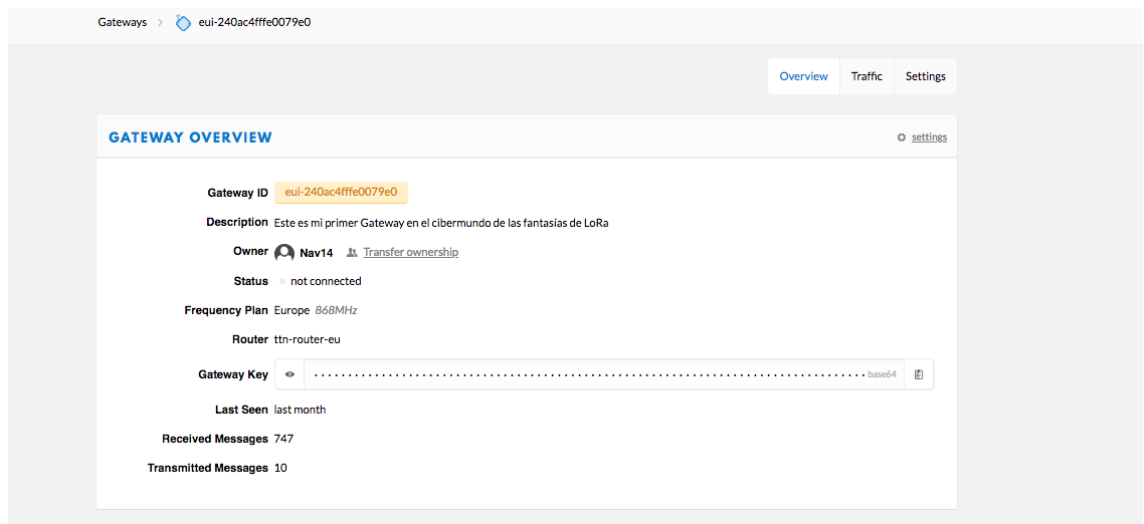


Figura 2.16 GW suscrito a TTN

Si se cumplen todos los requerimientos descritos nuestro dispositivo pasará a formar parte de la red TTN, y podremos acceder tanto a las características registradas de este como a una visualización en tiempo real del estado del dispositivo. En la Figura 2.16 puede verse un dispositivo registrado como Gateway, donde se visualiza haber asignado un Gtw ID, así como una clave para cumplir con los requerimientos de suscripción.

Cabe destacar que al igual que se ha de seguir un proceso de inscripción reglamentado para todo dispositivo que quiera formar parte de TTN. A su vez TTN estipula unas pautas respecto al uso de su red [8]. Dentro de las políticas de uso de su plataforma TTN estipula el uso de 30 segundos máximos en aire por día y por nodo para mensajes uplink, así como 10 mensajes downlink por día.

Por otro lado, estas restricciones pueden no ser necesarias, ya que TTN pone a disposición de sus usuarios su código fuente, para que puedan desarrollar su propia plataforma similar a TTN que conforme su total red privada de manera gratuita .[14]

2.3.1.2

Servicios

Todos los servicios que conforman el acceso y uso de TTN hacia la red LoRaWAN registrada se encuentra en la zona denominada como ‘consola’.

TTN conforma el back-end de la red (véase la Figura 2.2), lo cual concuerda con la subdivisión de la interfaz que encontramos en la plataforma. De acuerdo con el NS de la red encontramos el apartado ‘Gateways’, y conformando el AS encontramos la zona ‘Applications’.

En lo que respecta a la configuración e información disponible relativa al NS, TTN habilita dos funciones al usuario:

- Visualización de las transmisiones ocurridas a través del GW seleccionado.

- Configuración y visualización de las características que conforman el dispositivo, véase sus claves, así como la localización en la que se registró a este.

Respecto al AS, TTN proporciona una interfaz a través de la cual se acceden a las funciones de gestión de todo servicio relacionado con el nivel de aplicación de la red. Incluye funciones para la visualización del tráfico, así como la descripción de los dispositivos, pero en este caso, rigiendo los end-devices de la red.

Adicionalmente, incorpora dos servicios más a nivel de aplicación que permiten la interacción y transferencia de información de TTN con soluciones externas. Por un lado, ofrece una herramienta de decodificación, configurable por el usuario, que permite cambiar el formato del tráfico que se recibe. Este filtrado se aplica en una etapa previa a la exposición de los datos en el visualizador de la zona. En la Figura 2.17 se observa la interfaz proporcionada. Por otro, TTN proporciona un API que permite la interacción desde herramientas externas y generadas por el usuario. El conjunto de ambos servicios permite la generación de soluciones que aporten un valor añadido a la mera captura, transmisión y almacenamiento de datos. En el siguiente apartado se hablará en más profundidad sobre las integraciones existentes.



2.3.1.3

Figura 2.17 Interfaz de decodificación en TTN

Integraciones de servicios externos

TTN provee de una serie de integraciones entre las que destacan la inclusión de MQTT o HTTP, así como diversas APIs para otros lenguajes que permitan extender el sistema de su plataforma a otras redes. La Tabla 1 muestra un conjunto de las herramientas más comúnmente empleadas.

Tabla 1 Servicios externos a TTN

<u>Cayenne</u>	Herramienta que redirecciona los datos obtenidos de TTN a su propio servidor, en el cual poder tener los datos en tiempo real y almacenarlos.
<u>Collos</u>	Collos ofrece un conjunto de APIs gratuitas como la configuración de la red LoRa a una red con funcionamiento similar al WiFi, o un análisis del RSSI de manera constante en el sistema para que se autorregule en función a este.
<u>DataStorage</u>	Almacenaje del tráfico en TTN de la red registrada en un servidor gratuito. Se tiene una limitación de cantidad, así como solo un almacenamiento temporal de 7 días.
<u>HTTP Integration</u>	Integración para a redirección sobre http del tráfico de red en TTN a un dominio que se especifique. Habilita encaminar los datos a un servidor propio.
<u>OpenSensors</u>	Auto-denominado como ‘Twitter de sensores’ permite publicar todos los datos recibidos, funcionando de manera similar a una red social, también se puede acceder a los datos que el resto de usuarios publiquen.
<u>TTN Mapper</u>	Posicionamiento de los dispositivos registrados con TTN a través de compartir la posición GPS de un dispositivo móvil con el dispositivo LoRa.
<u>Tago</u>	Plataforma similar a TTN la cual permite el procesado, análisis y manejo del tráfico del sistema LoRaWAN.

2.3.2 TheThings.IO

TheThings.io [10] es una plataforma creada exclusivamente para trabajar sobre sistemas basados en múltiples protocolos.

Permite registrar e incorporar a su sistema cualquier Gateway que trabaje sobre LoRaWAN, o Gateways registrados en TTN. TheThings.IO es una plataforma en esencia similar a TTN, pero engloba en verdad un gran cúmulo de tecnologías como SigFox o integraciones sobre HTTP y MQTT. Su arquitectura se muestra en la Figura 2.18.

Las características que más destacan a la plataforma son:

- Herramientas de monitorización y visualización de datos
- Herramientas de análisis, procesamiento de datos.
- Integración de IA y autoaprendizaje a la red suscrita.
- Interoperabilidad con otras tecnologías e integraciones de diversos servicios.

La plataforma ofrece un periodo de prueba gratuito, tras lo cual acceder a ella pasa a ser un servicio de pago

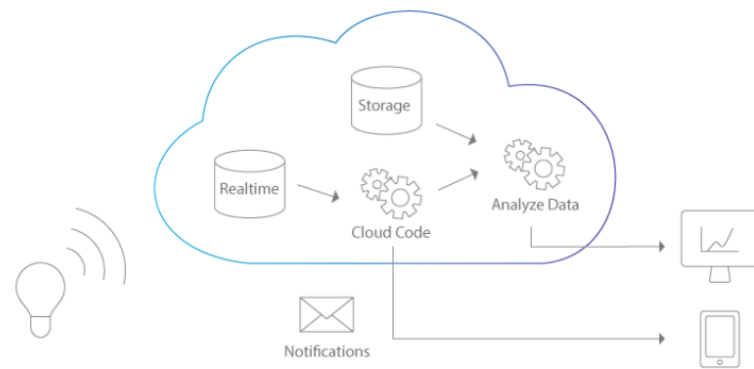


Figura 2.18 Arquitectura ofertada por TheThings.IO

2.3.3 ResIOT

ResIOT [9] es una plataforma de pago para la incorporación de sistemas IoT y sistemas basados en LoRaWAN. Proporciona una opción de uso gratuito si se dispone de un despliegue menor de 15 dispositivos activos. La Figura 2.19 muestra un ejemplo de arquitectura de despliegue y sus diferentes componentes.

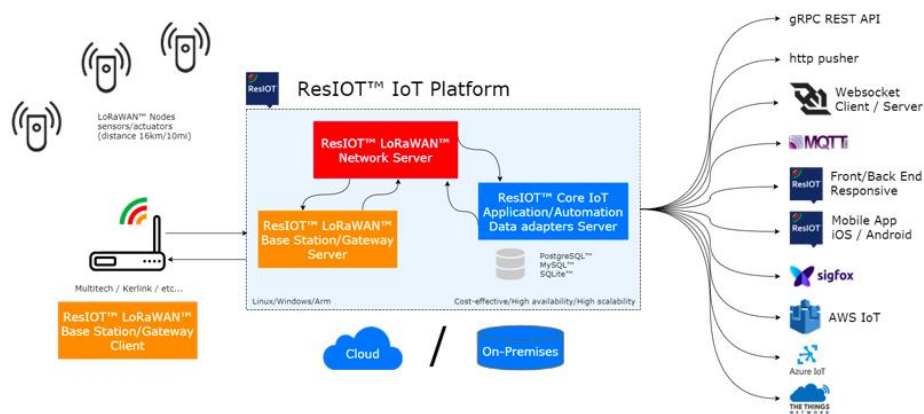


Figura 2.19 Arquitectura de red sobre Res.IO

ResIOT se encarga de la creación de la propia red, y sigue un patrón muy similar a la plataforma mencionada anteriormente. Ofrece al usuario:

- Gestión y configuración sobre los dispositivos de red
- Monitorización del sistema
- Agregar un nivel de seguridad basado en la plataforma
- Análisis y optimización de los componentes de la red
- Soporte para comunicaciones LoRaWAN en cualquier región.

3 Diseño y evaluación

Con el objetivo de evaluar la tecnología LoRa, se parte del uso de esta en exclusiva para el diseño de dos esquemas de comunicación sin recurrir a LoRaWAN para su desarrollo, encontrándonos así con dos modelos de comunicación basados en LoRa únicamente.

Este primer paso busca, una vez sean puestos en funcionamiento ambos sistemas, posibilitar la evaluación del rendimiento de estos y su caracterización, para posteriormente con el desarrollo conseguido, trabajar y constituir en el siguiente capítulo una solución global sobre LoRa que implemente LoRaWAN.

Inicialmente nos encontramos con sistemas que tienen una comunicación bruta entre dispositivos, sin seguir la estandarización marcada por el protocolo. Como evolución a estos, se procederá a implementar un sistema más organizado, dando ahora si roles a los dispositivos de la red, así como una topología de red. Así, se tiene ya no una comunicación bruta entre dos dispositivos, sino un sistema establecido, con su modo de transmisión y recepción organizado, asemejándose al desarrollo de un ‘nuevo’ protocolo.

Una vez establecido esta segunda red, podremos usar la experiencia de ambos desarrollos para meternos de lleno en el diseño de una red LoRaWAN completa.

3.1 Equipamiento hardware

Existen diversos dispositivos que incorporan la tecnología de comunicaciones LoRa empleando diversos módulos como el SX1276RF1IAS[12] o el módulo RN2483-I/RM101 [11], hasta diversas como la placa de Arduino MKR WAN 1300 [13].

No obstante, para el desarrollo de este proyecto y como primer acercamiento a la tecnología, se ha empleado el dispositivo LoPy de Pycom [15] por su versatilidad y la disponibilidad de un amplia documentación y comunidad que facilitará el desarrollo e integración de soluciones IoT. A su vez LoPy posee certificación en la plataforma TTN lo cual le permite operar dentro de ella, siendo este punto clave para el desarrollo del siguiente capítulo.

LoPy es un microcontrolador creado por Pycom enfocado a la integración de redes basadas en protocolo LoRa. LoPy es un componente muy versátil debido a su interoperabilidad entre las diversas varias tecnologías de comunicaciones que implementa, véase WiFi, SigFox, LoRa y Bluetooth.

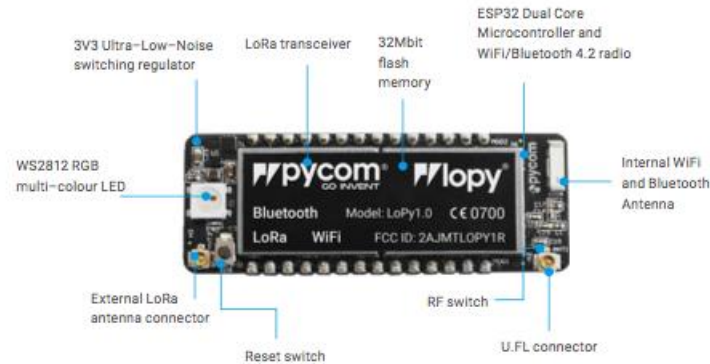


Figura 3.1 Hardware - placa LoPy

La Figura 3.1 muestra un ejemplo de un dispositivo LoPy. En la Tabla 2 se enumeran sus especificaciones más relevantes.

Tabla 2 Especificaciones LoPy

CPU	Xtensa dual-core 32-bit LX6 microprocessor(s)	
Memoria	RAM	Flash externa
	512 KB	4 MB
Wifi	802.11b/g/n	
Bluetooth	BLE	
LoRa	Implementa LoRaWAN posibilitando dispositivos de clase A y C	
RTC	150KHz	
Seguridad	Puede implementar SSL/TLS	

Como se ha descrito, LoPy aúna múltiples tecnologías de comunicación. No obstante, el diseño de la placa está enfocado como componente para generar una red LoRa. El resto de las tecnologías permiten extender las funcionalidades dentro de la red LoRa. Así, por ejemplo, la inclusión de la tecnología WiFi permite funcionar como Gateway y Bluetooth BLE habilita la generación de piconets de forma que se obtengan datos de sensores secundarios disponibles en el entorno.

El desarrollo de soluciones en LoPy emplea MycroPython[16], una versión de Python optimizada para correr en microcontroladores y sistemas embebidos de baja capacidad. La gestión de LoPy se realiza empleando, bien su propia consola de comandos conocida como REPL; habilitada a través de la extensión descargable para Atom Pymakr, a la que se puede acceder mediante una conexión UART directa, o bien mediante intercambios Telnet y FTP a través de la

interfaz WiFi operando en modo punto de acceso. En la Figura 3.2 Interfaz Atom y consola REPL podemos ver la disposición de Atom respecto a su consola REPL.

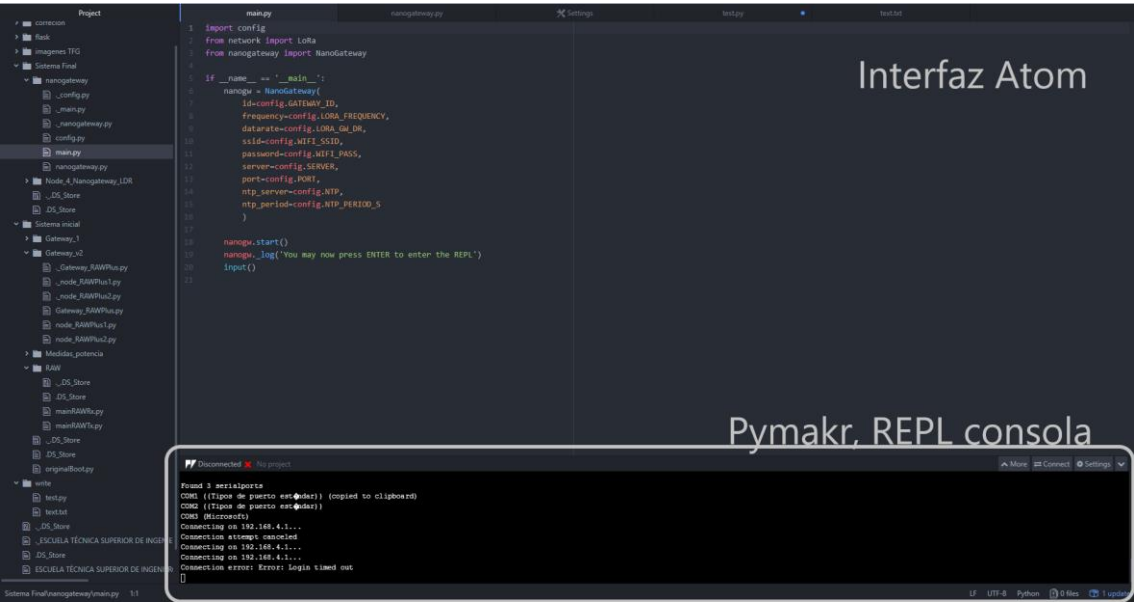


Figura 3.2 Interfaz Atom y consola REPL

Además de las tecnologías inalámbricas, LoPy permite interactuar con el exterior mediante los puertos GPIO digitales y analógicos que dispone. Esto ofrece una gran versatilidad a la hora de captar eventos externos debido a que la gama de sensores que pueden aplicarse sobre LoPy se ve más limitada por el número de pines de lectura que posee, antes que por aquellos tipos de sensores con los que es capaz de interactuar. En la Figura 3.3 se puede ver la exposición del pinout que compone al dispositivo LoPy.

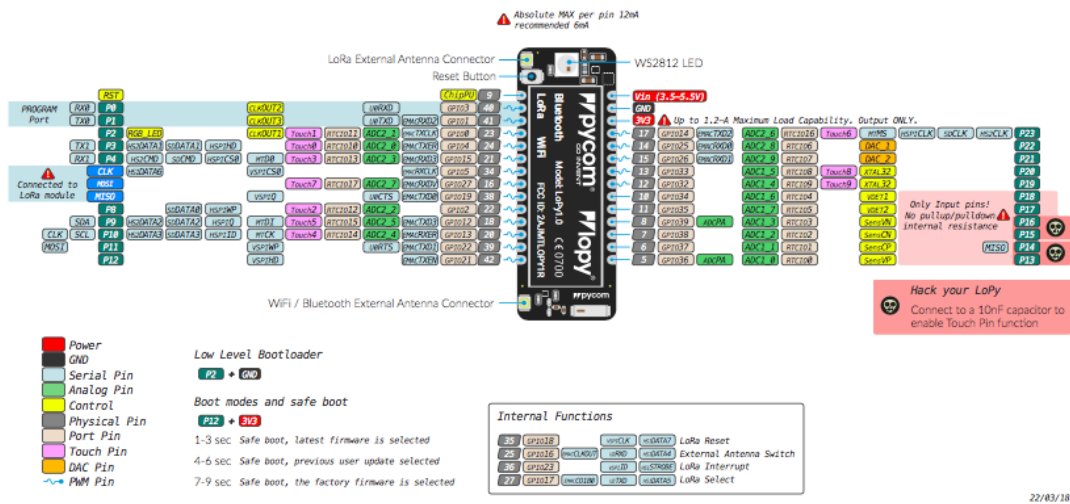


Figura 3.3 Pinout LoPy

PyCom dispone de diversas placas de expansión para LoPy de forma que se le puedan añadir más funcionalidades e interfaces de entrada/salida. La placa Expansion board 3.0 [15], mostrada en

la Figura 3.4, ofrece junto con el conjunto de interfaces I/O extra una ranura para tarjeta microSD en la que poder almacenar cualquier tipo de información, desde datos capturados hasta las propias aplicaciones. El diseño de esta placa está enfocado en ofrecer al usuario la capacidad de insertar gran variedad de sensores, y usar una tarjeta SD como suplemento para poder soportar el almacenamiento de los datos que esto conllevaría.

Otra opción es el uso de PyTrack [15] (Figura 3.5) que incorpora un módulo GPS y acelerómetro, con el que poder determinar la posición de los dispositivos en tiempo real. Para evitar consumos excesivos, PyTrack incorpora un modo de operación a potencia mínima o modo de sleep.

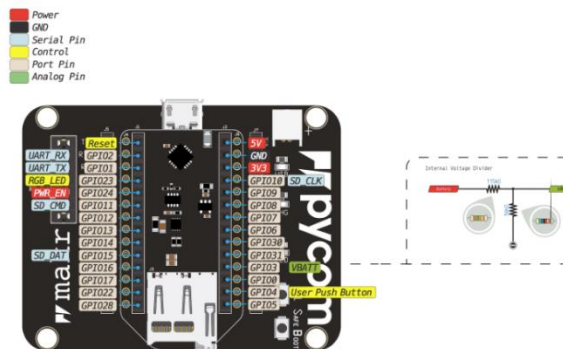


Figura 3.4 Expansion Board 3.0

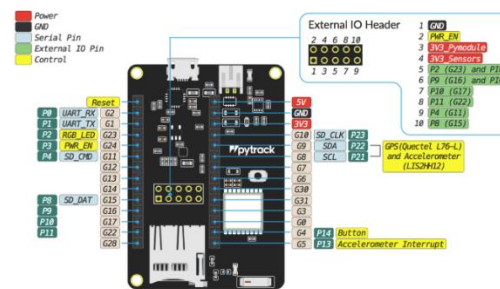


Figura 3.5 PyTrack

Para el presente proyecto la configuración que se aplicará estará constituida esencialmente por:

- Módulo Lopy
- Placa de expansión PyTrack para obtener información de posicionamiento
- Placa de expansión 3.0 para conectar sensores.
- Una antena conectada a su conector LoRa/WiFi.

En esta primera etapa de diseño, se recurrirá al uso de la placa de expansión PyTrack como se expondrá en el apartado siguiente.

3.2 Red LoRaRAW

El primer sistema a evaluar emplea la arquitectura más simple y limpia en todos los niveles de la pila de protocolos. El diseño del sistema lo conforman dos elementos, un nodo transmisor y uno receptor. En este sentido para caracterizar el sistema de comunicaciones, la aproximación que se ha realizado es emplear una configuración de comunicaciones en bruto:

- No recurre a nivel de enlace.
- No aplique roles en la red.
- Topología no estructurada.

LoRaRAW [17] es método que ofrece Pycom en sus librerías para implementar una comunicación en bruto entre cualesquiera dispositivos que implementen este método y estén activos. Este modo de trabajo no es recomendable para el uso en redes en producción.

LoRaRAW no implementa seguridad alguna, y la información se envía a todo aquel que implementado el método este a la escucha, es decir, sea o no parte del sistema pero que implemente LoRaRAW.

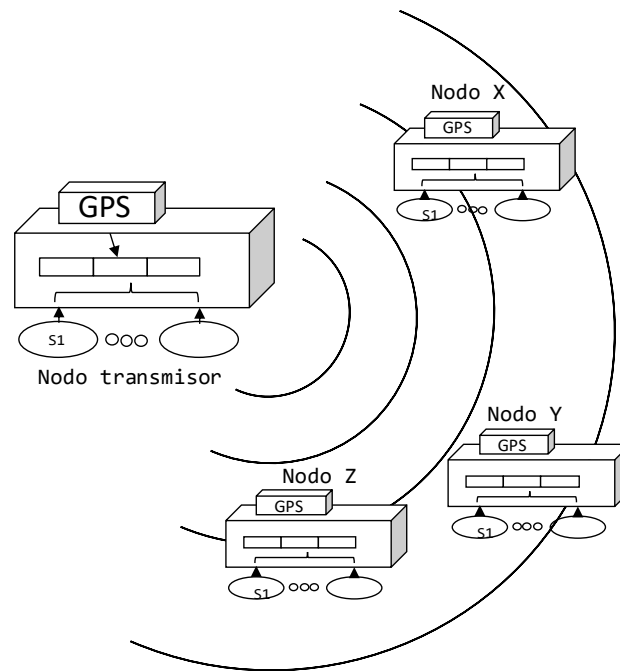


Figura 3.6 Transmisión LoRaRAW

En la Figura 3.6, se muestra la disposición del despliegue realizado. Cada nodo está formado por una placa LoPy a la que se le ha conectado la placa PyTrack que ofrece su GPS. Cada mensaje que transmite un nodo incluye su posición GPS. De este modo, será posible determinar la distancia aproximada con el nodo receptor y caracterizar así la cobertura.

Aunque el transmisor trate de enviar el mensaje solo a un nodo, LoRaRAW únicamente implementa la difusión de mensajes (transmisión broadcast), por lo que cualquier otro nodo en modo escucha dentro del rango de cobertura recibirá el mensaje.

LoRaRAW emplea una metodología de acceso al medio muy similar a las comunicaciones Aloha [18] en sus primeras etapas con Aloha puro, pues se tiene un acceso totalmente aleatorio a la red y los nodos desconocen el estado del resto de dispositivos de esta. Es por ello que no se considera que haya topología en la red.

Para el caso que ocupa, no se han implementado retransmisiones, por la sobrecarga adicional. Es por esto que cualquier paquete fuera de rango no será recibido, pudiendo considerarse en la recepción que la red está caída o no operativa.

3.2.1 Configuración de los dispositivos

Se procede a detallar como se ha programado los dispositivos para la transmisión y recepción. Tanto transmisor como receptor se han personalizado con su correspondiente aplicación: por una parte, el transmisor realiza transmisiones de forma periódica de su posición, mientras que el receptor recupera dicho mensaje y lo almacena con estadísticas relativas a las características de la recepción que permitirán posteriormente evaluar el rendimiento.

Desde el punto de vista software ambos nodos incluyen:

- Main.py: programa principal de transmisor/receptor.
- L76GNSS.py: librería para el uso del módulo GPS.
- Pyrcoproc.py: librería para configuración de placa PyTrack.
- Pytrack.py: inicializador de la configuración de la placa PyTrack.

Transmisor

3.2.1.1

El programa principal que ejecuta el transmisor se rige por el diagrama de estados que se incluye en la Figura 3.7.

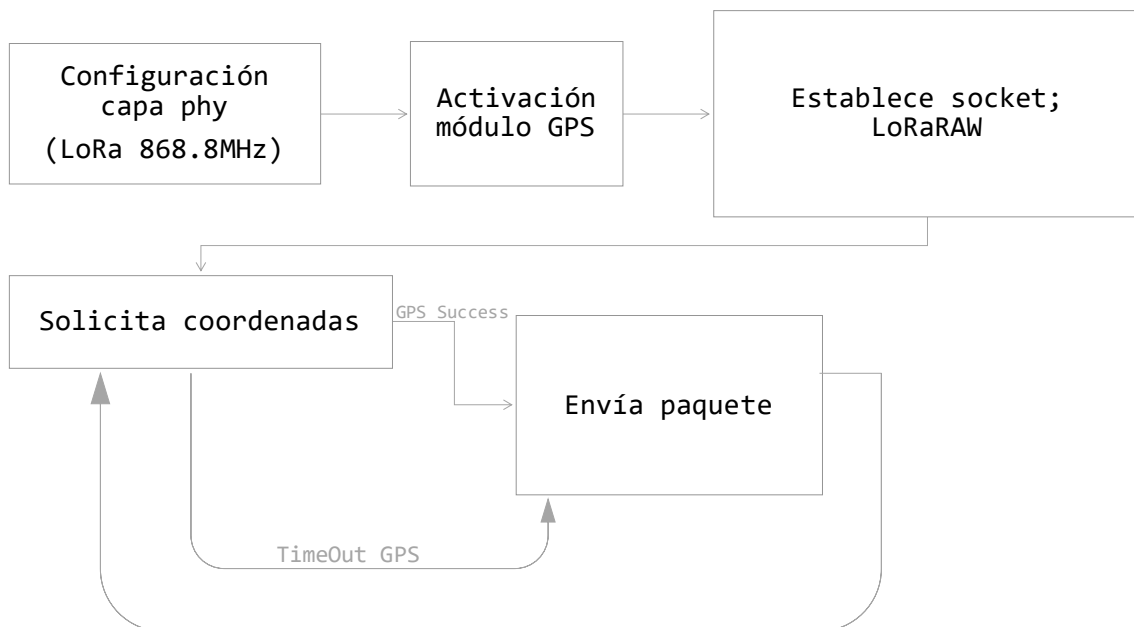


Figura 3.7 Proceso transmisión-Sistema LoRaRAW

A modo de ejemplo del código empleado en la programación, para la aplicar la configuración del modo de comunicaciones de la placa LoPy se emplea la siguiente instrucción:

```
LoRa(mode=LoRa."LORA",region="LoRa.EU868")
```

Este procedimiento de inicialización es común a todo dispositivo de los diversos sistemas LoRa que se van a emplear y es aquel que nos permite detallar si el método a ejecutar es LoRaRAW o LoRaWAN. Respecto al parámetro región, se rige función a las bandas ISM de cada región descritas en 2.1, siendo para nuestro caso la frecuencia utilizada de 868Mhz.

El módulo GPS permanece activado desde el inicio del dispositivo. Una vez se activa comienza a sincronizarse con los satélites disponibles y obtiene los valores relativos a la posición. Para la obtención de las coordenadas, se consideran dos estados:

- Si se tiene una buena recepción, se recibe la latitud y longitud, se considerará la transición al estado GPS Success, conformando el paquete y procediendo a su transmisión.
- Si pasado un tiempo de espera establecido, no se ha conseguido obtener las coordenadas, véase porque el dispositivo acaba de encenderse o se ha perdido la visión directa para la comunicación satelital, se devuelven, los valores (None,None) representando posición desconocida, transmitiéndose posteriormente un paquete con dicha información.

Este proceso de petición de posición GPS, más transmisión se procede a ejecutar de manera constante en un bucle, con un periodo de espera establecido entre transmisiones.

3.2.1.2

Receptor

El proceso de recepción sigue un procedimiento de inicialización similar al del transmisor. Puesto que el dispositivo receptor se mantendrá en un lugar fijo, no requiere de ir obteniendo sus coordenadas GPS, pues estas no variarán. El nodo receptor, si bien en la configuración definida no existen roles, actúa como el equivalente a un GW que recibe los datos de los nodos de su entorno, se mantiene una vez configurado en un proceso de recepción constante hasta obtener resultados. Este proceso puede verse representado en la Figura 3.8.

Ya que se hará una caracterización de los resultados obtenidos de múltiples transmisiones, se requiere de un archivo donde poder ir guardando los resultados. El receptor sistema mantendrá un archivo de log, donde se almacenará cada paquete recibido y las características de la transmisión/recepción.

El proceso de recepción, al igual que en la transmisión, es un proceso cíclico donde el dispositivo permanece en un estado de escucha permanente. Si le llega algún paquete; inmediatamente lo recoge y almacena las características y estadísticas asociadas a él. Esto se consigue mediante el método `stats()`[17] que devuelve un vector conformado por las diversas características de la última transmisión que se haya recibido.

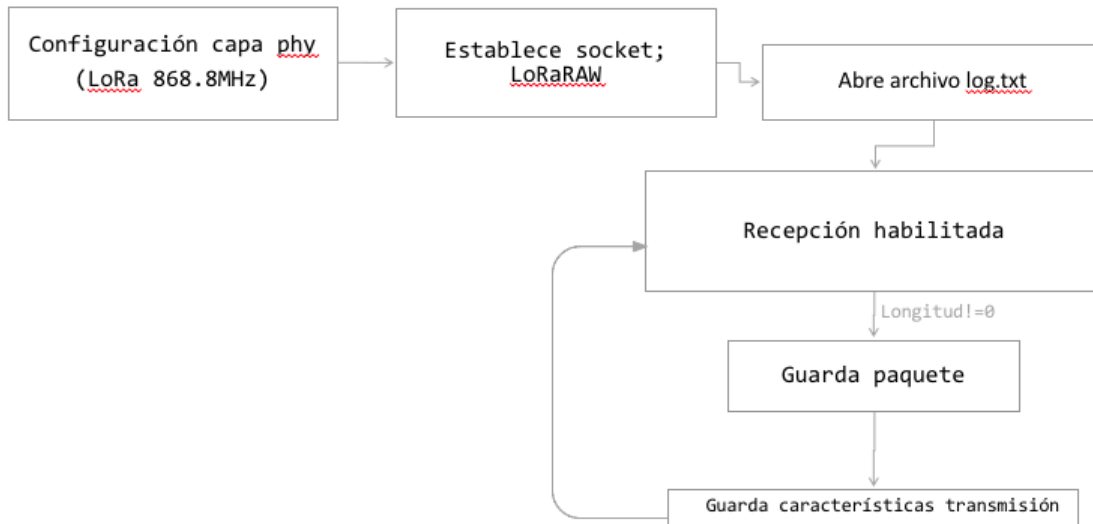


Figura 3.8 Recepción LoRaRAW

3.2.2 Caracterización del sistema

Se disponen dos nodos, uno transmisor y otro receptor, que intercambian un conjunto de mensajes.

Con el diseño actual no podemos observar la cantidad de fallos de transmisión que ocurren, pero si se puede obtener información de las estadísticas asociadas a cada paquete transmitido.

- Posición GPS
- Received signal strength indication (RSSI): escala de referencia para medir el nivel de potencia de las señales recibidas por un dispositivo en las redes inalámbricas
- Signal to Noise Ratio (SNR): proporción entre la potencia transmitida y la interferencia causada por el ruido en esta.
- Spreading Factor: parámetro configurable en las comunicaciones LoRa para la variación del área de cobertura de la red. Su funcionamiento se ve detallado en Figura 2.1.

Se procede a hacer un primer test de la tecnología LoRa en este sistema para poner a prueba la cobertura que este realmente ofrece. En función a las coordenadas GPS podremos obtener la distancia entre los dispositivos emisor y receptor y analizando los valores del RSSI junto a la SNR de la señal, podremos evaluar la evolución de la calidad de recepción en función de la distancia.

Esta primera prueba se realiza en una zona donde se pueda obtener línea de visión directa entre el emisor y el receptor. Para realizar las mediciones se han posicionado los nodos en el parque el Parque de Las Llamas en la localidad de Santander, según se muestra en la Figura 3.9 .

En esta medición inicial se cubrirá una zona con línea de visión directa entre transmisor y receptor (fijado en la posición de referencia A). Se irán tomando medidas intermedias en 5

posiciones cada 200 m, desde las cuales se realizarán 5 transmisiones. En la **¡Error! No se encuentra el origen de la referencia.** se muestran dichas posiciones.

Tabla 3 Despliegue caracterización sistema LoRaRAW

Localización	Latitud	Longitud	Distancia al GW
A-GW	43.475333	-3.797118	0m.
B-Nodo en P1	43.474749	-3.799017	200m.
C-Nodo en P2	43.474103	-3.800959	400m.
D-Nodo en P3	43.473651	-3.803780	600m.
E-Nodo en P4	43.472896	-3.806366	800m.
F-Nodo en P5	43.47.20.47	-3.808222	1000m.



Figura 3.9 Puntos de medición en caracterización sistema LoRaRAW

Tras realizar todo el proceso considerando una potencia de TX de 14 dBm y un spread factor de 7. El valor de SF a 7 se selecciona dado que es el menor valor posible y con el que se obtiene una sensibilidad de se tiene una sensibilidad entre -120dBm y -123dBm en el receptor y el menor rango de cobertura.

Para la exposición de los resultados, se procede a mediar los resultados obtenidos, y ofrecer una visión más completa complementado este valor medio con el máximo y mínimo obtenido en cada uno de los parámetros como puede verse en la Tabla 4 -Media resultados caracterización sistema LoRaRAW Tabla 4 a continuación.

Tabla 4 -Media resultados caracterización sistema LoRaRAW

Posición	RSSI (dBm)			SNR		
	Mínimo	Medio	Máximo	Mínimo	Medio	Máximo
B	-79	-89.4	-95	4.0	5.8	7.0
C	-102	-105.8	-109	5.0	5.8	7.0
D	-103	-104	-105	5.0	5.8	7.0
E	-114	-119.2	-123	3.0	4	6.0
F	-108	-110.6	-115	2.0	4.8	6.0

Con los resultados obtenidos, a modo de mejor perspectiva se procede al desarrollo de gráficas que expongan los resultados para su posterior análisis como pueden verse en las gráficas correspondientes a la Figura 3.10.

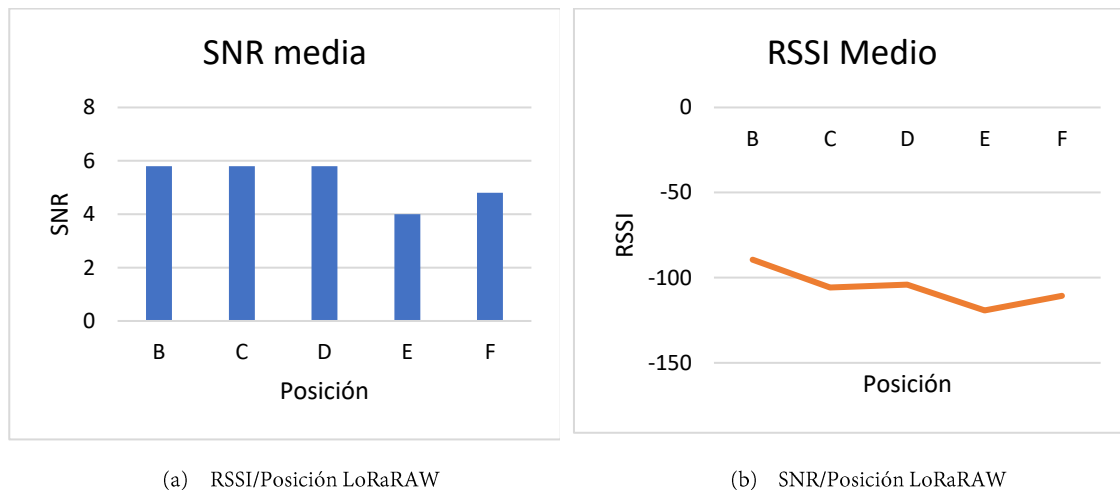


Figura 3.10 Caracterización sistema LoRaRAW

Todos los paquetes enviados han llegado sin fallo, y a lo largo del proceso se observa que, a pesar de un aumento considerable, de un 1Km en la distancia, se mantienen parámetros aceptables en el RSSI recibido.

Respecto a la calidad de la señal, se observa un deterioro en la medida en el punto E a 800 m, que significaría el comienzo de la pérdida en la calidad. Sin embargo, en F a 1 Km, se obtienen mejores resultados. Esto se puede deber posiblemente a los obstáculos encontrados entre el punto de transmisión y recepción en el punto E, mientras que el punto F, aunque se posicione a mayor distancia, posee una mejor línea de visión sin tantos obstáculos respecto al punto A.

Se puede ver un deterioro ligero en la recepción, pero este deterioro se ve desde el punto B a 200 metros, al último lugar de medición. Por ello se puede concluir que se mantiene estable tanto el SNR como el RSSI medido entre 200 metros a 1 Km, con un mayor deterioro en la SNR.

De este análisis se obtienen los límites de cobertura para el sistema sin gestión de nivel MAC en las transmisiones. Será con el segundo procedimiento de caracterización donde además de incluir retransmisiones, se evalúe también la influencia de parámetros como el SF en la variación del radio de cobertura, tasa binaria, etc.

Ha de mencionarse que este despliegue se ha tratado de implementar en el medio más favorable posible, tratando en todo momento de otorgar la visión más optimista que podría obtenerse con un sistema LoRa.

Por otro lado, la restricción a 1 Km de las medidas no se debe solo a querer caracterizar esta área de cobertura, pues se ha tratado de hacer un incremento de cobertura respecto a esta distancia en las pruebas realizadas en el entorno, solo llevando a la pérdida de comunicación entre transmisor y receptor en cuanto se procedía a un incremento de esta distancia, por una gran pérdida de línea de visión entre transmisor y receptor.

3.3 Sistema LoRa-MAC, nodo-Gateway

Tomando como base el despliegue del apartado anterior, se busca hacer una serie de mejoras en el sistema. Se aplicará ahora el modelo de transmisión LoRa-MAC [17], como evolución de LoRaRAW, y cuyas características principales consisten en:

- Añadir roles a la red
- Añadir retransmisiones
- Capacidad de detectar fallos de transmisión
- Establecer un formato de paquete
- Envíos destinados y confirmados

Se ha de clarificar que LoRa-MAC no es un método nuevo como tal, sino la configuración de un modo de acceso al medio añadido por el diseñador de la red trabajando sobre el método LoRaRAW. Por tanto, se basa en construir un protocolo sobre las comunicaciones brutas que se han implementado en el anterior sistema.

El modo de funcionamiento, modo de acceso, y características de LoRa-MAC son seleccionadas por el diseñador. Por ello cada sistema LoRa-MAC hace referencia a un protocolo diseñado propiamente solo para este sistema, pudiendo cada persona que desea desplegar una red basada en LoRa-MAC tener un modelo de acceso al medio, y comportamiento en la red totalmente diferente.

Se procede a generar dos nodos transmisores, los cuales envían a un sumidero, representado por otro nodo con la función de GW. Dicho nodo recogerá los mensajes enviados por los dos nodos

transmisores, los cuales tendrán un formato preestablecido y decidirá a cuál de ambos nodos ha de reenviar un mensaje de confirmación o ACK. Consiguiendo así un modelo de comunicación confirmada el cual a la hora de evaluar la cobertura de la red como se hará más adelante, nos permita ver en qué momento se pierde la cobertura, por la no recepción del ACK y el intento de retransmisión.

En la Figura 3.11 Esquema sistema LoRaMAC se muestra esquemáticamente el funcionamiento de este modo de comunicación. Teniendo los dos transmisores mencionados; coincidiendo su hardware con el de los transmisores del sistema LoRaRAW, emitiendo paquetes hacia el GW de la red. Respondiendo este una vez recibido el paquete con la confirmación hacia el transmisor del que procede el paquete.

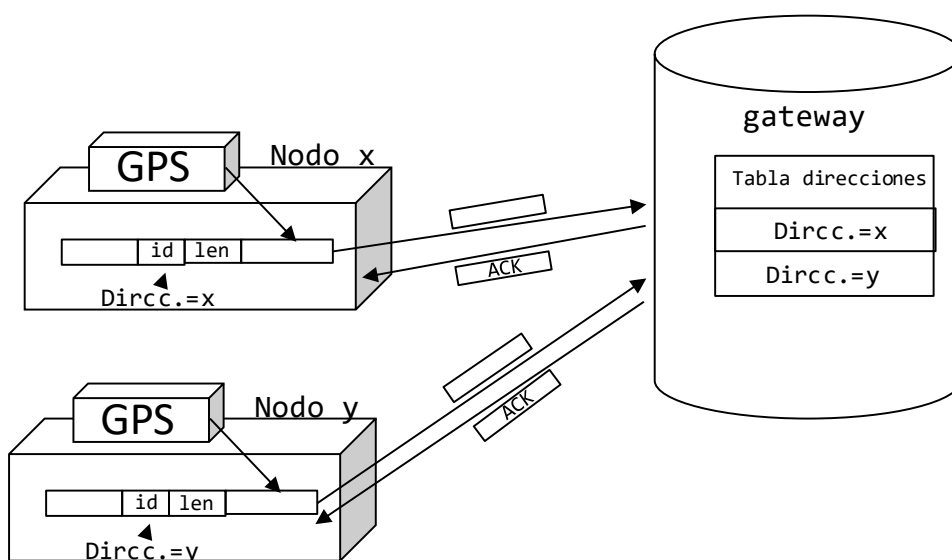


Figura 3.11 Esquema sistema LoRaMAC

3.3.1 Configuración de los dispositivos

El proceso de transmisión y recepción procede de una manera muy similar al descrito en la sección anterior, si bien en este caso, se incluye la potencial retransmisión de mensajes en caso de no recibir una confirmación de recepción.

Como tal la transmisión se hace desde el método LoRaRAW, pero sobre este se construye un modelo el cual permita que los nodos transmisores envíen sus paquetes no entre ellos, sino hacia el sumidero de la red; el GW, tratando de acercarnos hacia la topología que se nos muestra en la Figura 2.2.

Por su parte el nodo transmisor ya no ejerce solo un papel pasivo de recepción, sino que forma activamente parte de la red con la confirmación de las transmisiones recibidas.

Transmisor

Tal como se muestra en la Figura 3.12, se procede de forma similar a la configuración del dispositivo y a la activación del módulo GPS. Tras obtener las coordenadas y se conforma el paquete y se procede a la transmisión.

3.3.1.1

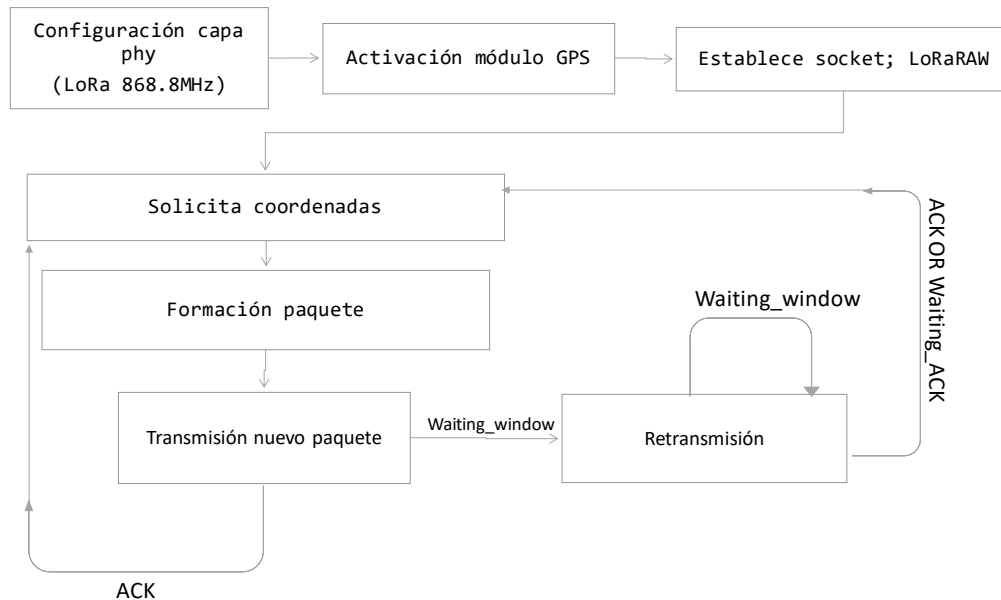


Figura 3.12 Transmisión sistema LoRaMAC

Una vez un paquete es transmitido se espera a la recepción de una confirmación ACK por parte del GW. En caso de no recibirse, se aplican nuevas opciones en la máquina de estados en función de dos parámetros:

- **Waiting_window**: es el tiempo de espera por parte del nodo entre retransmisiones si no recibe un ACK.
- **ACK_window**: tiempo máximo en el cual procede a retransmitir el paquete antes del descarte de este.

En este nuevo escenario se definen 3 estados para una transmisión:

- Comunicación exitosa.
- Comunicación exitosa con retransmisión, requiere de haber pasado varias **Waiting_window**, sin rebasar el tiempo del **ACK_window**.
- Comunicación fallida, se descarta el paquete tras el **ACK_window** y se procede a transmitir otro nuevo.

Receptor

El proceso de recepción incorpora dos etapas con respecto a la versión inicial del receptor. Por un lado, debe descifrar el paquete y obtener el identificador del dispositivo transmisor y por otro generar un paquete de confirmación y redirigirlo al transmisor. Se mantienen iguales las rutinas de almacenaje de paquetes y características de la transmisión. En la Figura 3.13 se muestra el procedimiento descrito.

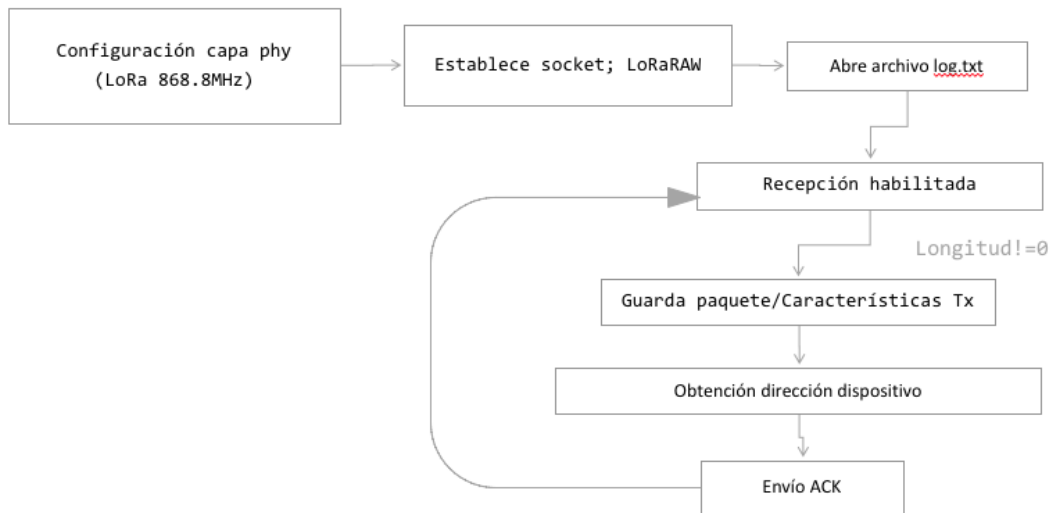


Figura 3.13 Recepción sistema LoRa-MAC

3.3.1.3

Formato de paquetes:

Este nuevo sistema requiere de añadir un formato de paquete para hacer la comunicación entre dispositivos. En la Figura 3.14 y en la Figura 3.15 se muestran los formatos del paquete de información transmitidos en el sentido Nodo a GW y el de confirmación en el sentido GW-nodo.

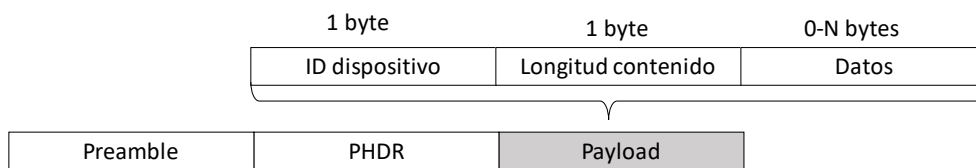


Figura 3.14 Formato datos LoRa-MAC

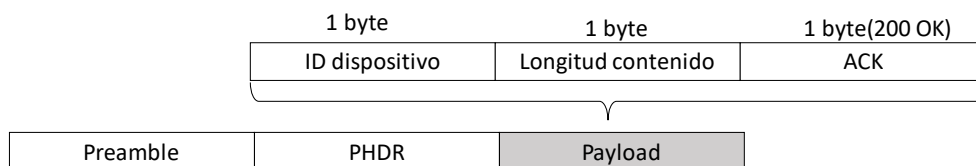


Figura 3.15 Formato ACK LoRa-MAC

Los identificadores de dispositivo se preestablecen en cada dispositivo durante el proceso de configuración. De esta forma, es posible direccionar correctamente la información en ambos sentidos, posibilitando descartar paquetes cuyo destino no es el correcto.

Tanto el preámbulo como la cabecera PHDR se mantiene al igual que en el sistema LoRaRAW al estar implementándose sobre LoRaRAW y por tanto si estar implementando la capa física que constituye LoRa en LoRaWAN.

De esta cabecera física es de la que podemos obtener los parámetros obtenidos a través de nuevo del uso del método `stats()`.

3.3.2 Caracterización del sistema

Para este caso se va a hacer dos tomas de medida diferentes. En primer lugar, se repetirá una campaña de medidas similar a la realizada para el caso de LoRaRAW. En segundo lugar, se realizará una caracterización variando el SF para determinar su influencia.

Sistema LoRa-MAC

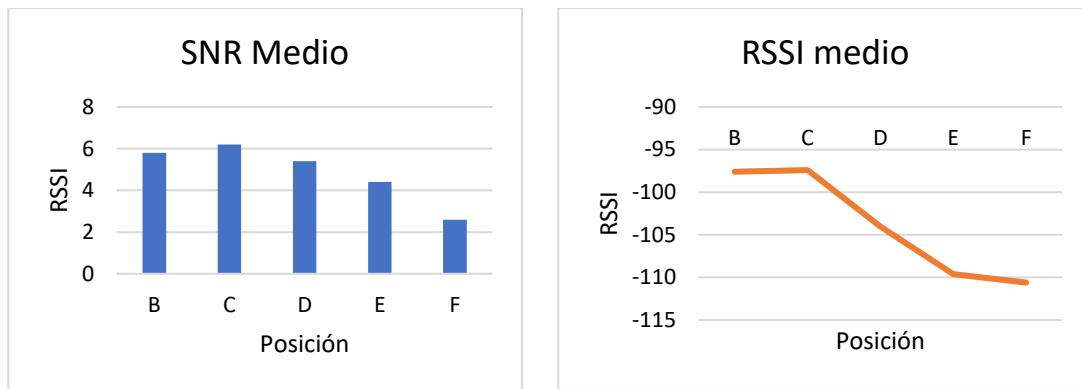
3.3.2.1 La modificación en la capa de acceso al medio y el modo de operación de los nodos permitirá obtener más información que en el caso anterior. Ahora se podrá disponer de valores como el número de transmisiones o el tiempo en el aire o latencia del paquete.

La Tabla 5 muestra el resumen de la campaña de medidas realizadas en las cinco posiciones de la Figura 3.9.

Tabla 5 Resultados 1ª campaña de medida LoRa-MAC

Posición	RSSI (dBm)			SNR		
	Mínimo	Medio	Máximo	Mínimo	Medio	Máximo
B	-94	-97.6	-102	5.0	5.8	6.0
C	-94	-97.4	-102	6.0	6.2	7.0
D	-100	-104	-108	4.0	5.4	6.0
E	-108	-109.6	-112	3.0	4.4	5.0
F	-109	-110.6	-113	2.0	2.6	3.0

Al igual que en la campaña de LoRaRAW se procede a graficar los resultados como puede observarse en la Figura 3.16.



(a) RSSI/Posición 1ª medición

(b) SNR/Posición 1ª medición

Figura 3.16 Caracterización LoRa-MAC en primera medición

Como se puede observar, al igual que en la anterior campaña similar a esta, se observa un deterioro en la calidad de la señal según vamos alejándonos del punto A.

En este caso la campaña desarrollada muestra un comportamiento menos impredecible, viéndose como en el punto E decrece tanto el RSSI como la SNR para tener las peores calidades de recepción en F, a diferencia de la mejora que se observaba en la Figura 3.10 que se corresponde a los resultados de la anterior campaña.

Por otro lado, sí que se mantiene la pérdida de comunicación entre transmisor y GW una vez se supera 1Km de distancia, momento en el cual procede a retransmitir el transmisor hasta completar su Ack_Window, tras lo cual procede a entrar en bucle en la generación de nuevos paquetes y reintento de retransmitir sin éxito.

3.3.2.2

Variación del SF

Para la segunda prueba de cobertura realizada, se ha buscado caracterizar la red en función del SF. La idea es realizar tres campañas de medidas en las que se aumente el SF, buscando el punto a partir del cual incurran en errores de transmisión o pérdidas de mensajes, para poder determinar en cada caso el rango operativo de transmisión del sistema.

Se realizarán medidas con valores de SF de 7,9 y 12. Se mantendrá fijo un GW en el exterior de un laboratorio del Grupo de Ingeniería Telemática en la Escuela Superior de Ingenieros Industriales y de Telecomunicación de la Universidad de Cantabria (43.47302, -3.798038), desde el cual se tenga una línea de visión hacia la zona donde se realizaron las medidas del sistema anterior.

El transmisor deberá recibir y visualizar el mensaje ACK de vuelta por parte del GW si la recepción aún funciona. En el punto en que ya no se capten más ACK por parte del GW, se dará por hecho por tanto que se ha llegado a ese horizonte de comunicación y se guardaran todos los datos obtenidos para su estudio posterior.

Como en la campaña anterior, se mantiene una potencia de transmisión de 14 dBm en las transmisiones.

- SF 7

Tabla 6 Campaña de medición son SF variable, SF=7

Distancia	SNR	RSSI
0-100	5,66666667	-94,666667
100-200	6	-91
200-300	6,09090909	-82,727273
300-400	6,14285714	-86
400-500	5,8	-93,4
500-600	6,07142857	-91,285714

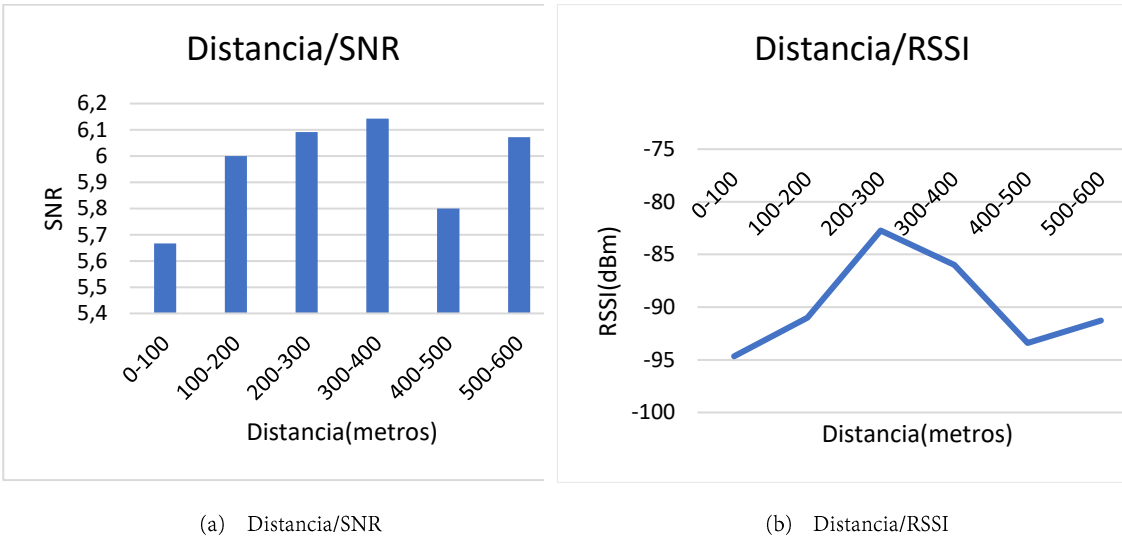
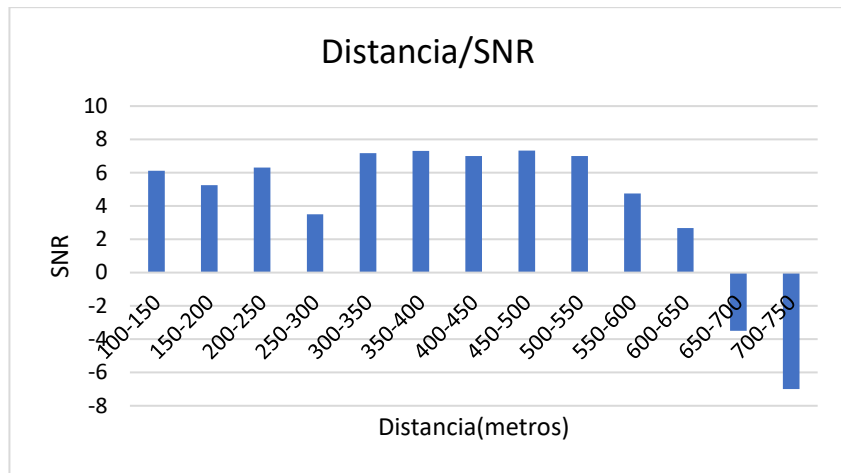


Figura 3.17 Campaña de medición con SF variable, SF=7

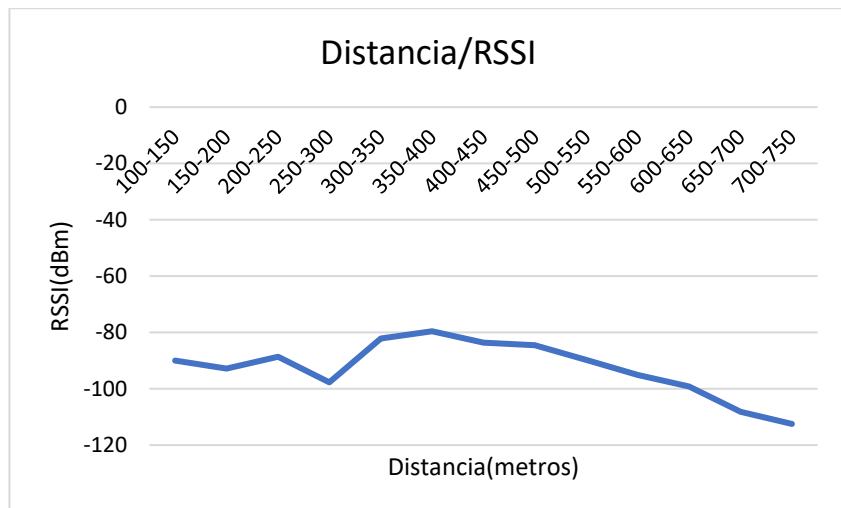
- SF 9

Tabla 7 Campaña de medición son SF variable, SF=9

Distancia	SNR	RSSI
100-150	6,125	-90
150-200	5,25	-92,875
200-250	6,3	-88,6
250-300	3,5	-97,75
300-350	7,18181818	-82,181818
350-400	7,3	-79,6
400-450	7	-83,666667
450-500	7,33333333	-84,555556
500-550	7	-89,75
550-600	4,75	-95,125
600-650	2,66666667	-99,166667
650-700	-3,5	-108,16667



(a) Distancia/SNR



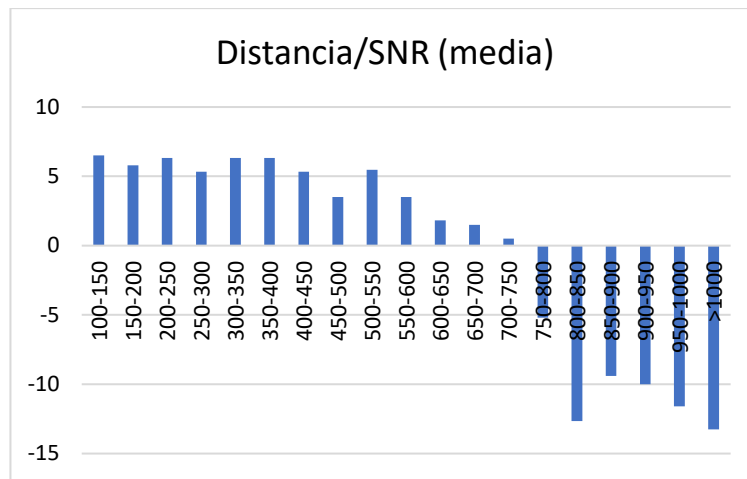
(b) Distancia/RSSI

Figura 3.18 Campaña de medición son SF variable, SF=9

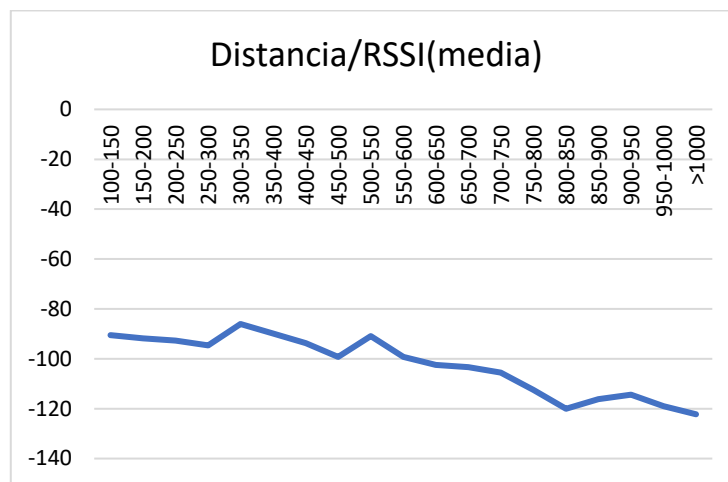
- SF 12

Tabla 8 Campaña de medición son SF variable, SF=12

Distancia	SNR	RSSI
100-150	6,5	-90,5
150-200	5,8	-91,8
200-250	6,33333333	-92,666667
250-300	5,33333333	-94,666667
300-350	6,31578947	-86,052632
350-400	6,33333333	-89,888889
400-450	5,33333333	-93,75
450-500	3,5	-99,25
500-550	5,47368421	-90,842105
550-600	3,5	-99,214286
600-650	1,83333333	-102,5
650-700	1,5	-103,33333
700-750	0,5	-105,5
750-800	-5,2	-112,44
800-850	-12,666667	-120
850-900	-9,4	-116,2
900-950	-10	-114,33333
950-1000	-11,6	-119



(a) Distancia/SNR



(a) Distancia/RSSI

Figura 3.19 Campaña de medición con SF variable, SF=12

Los resultados obtenidos por un lado corroboran que la configuración del SF en las comunicaciones LoRa permite configurar la cobertura ofrecida por una red LoRa. Se han obtenido 3 resultados diferentes (Tabla 6, Tabla 7 y Tabla 8), con el SF mínimo con valor 7, donde solo se ha llegado a una distancia de 600 metros, yendo está en aumento en las posteriores campañas con el SF a 9 y 12.

Cabe destacar la gran diferencia que se puede observar respecto al resultado obtenido en las dos campañas anteriores que con el mismo valor de SF cubrían 1 Km. Se asume que esta pérdida de cobertura se debe al nuevo entorno sobre el que se dispone el sistema, ya que se ha comprobado la pérdida de comunicación a partir de los 600 metros. Aunque en la Figura 3.17 se observa cómo entre los 400 y 500 metros tenemos los peores resultados respecto a la recepción, al acercarnos a los 600 metros se observa una ligera mejora. No con ello, se consigue que a partir de estos 600 metros se llegué a quedarnos en un constante bucle de transmisión sin respuesta por parte de la recepción, por lo que se presupone este como el horizonte de cobertura.

En segundo lugar, se consigue aumentar en 150 metros la cobertura del sistema (Figura 3.18), colocando un SF de 9 en la segunda medición, pasando el horizonte de cobertura anterior es cuando podemos comenzar a apreciar un gran deterioro en los resultados, pero al haber procurado un aumento en el SF de la comunicación, se consigue mantener cobertura hasta los 750 metros, donde nos encontramos de nuevo con el siguiente horizonte de cobertura.

Por último, se cubre todo 1Km de cobertura con el SF a su máximo valor 12(Figura 3.19). En esta última medición se aprecia un gran deterioro en las señales recibidas una vez superamos los 750 metros límite, pero debido al aumento del SF a pesar de la disminución del RSSI y SNR en las señales que captamos, siguen siendo captadas. Hasta llegar a un progresivo deterioro en la recepción perdiendo la comunicación al llegar al kilómetro de distancia.

Nos encontramos así un proceso recursivo, llegamos a un límite de cobertura, se aumenta el SF, conseguimos un límite de cobertura mayor. Por ello se puede considerar que el aumento del SF conlleva un aumento de nuestro horizonte de comunicación.

En este segundo entorno en que las medidas se han realizado se ha dispuesto el GW de un modo más realista, colocándolo en una posición más elevada que lo favorezca a la hora de recibir las transmisiones. Pero los impedimentos que iban surgiendo a medida que se aumentaba la distancia son más realistas que en el primer sistema, mayor cantidad de obstáculos e impedimentos para la línea de visión entre nodo y GW.

4 Sistema LoRaWAN

En este capítulo se pretende emplear los conocimientos adquiridos para diseñar, desarrollar y desplegar un sistema basado en LoRaWAN completo y funcional. Si bien el capítulo anterior se ha focalizado en el análisis del rendimiento de la tecnología LoRa, este se centrará en su uso en la provisión de servicios IoT completo, desde la obtención de datos, su transporte, su almacenaje y posterior procesamiento y presentación al usuario procesado.

La plataforma por desplegar seguirá la especificación LoRaWAN [6]. Por ello se tendrá una zona conformada por los nodos sensores, un GW a modo de sumidero LoRa que redirigirá toda la información a un AS y un NS.

Se plantea también la integración en una plataforma LoRaWAN comercial, como es TTN que ofrece un servicio propio de almacén de los datos recolectados por la red LoRa, así como el desarrollo de un sencillo interfaz de usuario externo.

Para finalizar, una tecnología requiere de ser usada en un caso más práctico y realista, haciendo uso tanto de TTN como sus herramientas, y la creación de un servidor propio.

La idea en esta última etapa es crear un sistema completo donde tengamos una zona inicial de recogida de datos sensoriales, TTN como NS y AS, y un back-end conformado por este servidor desarrollado con Flask en conjunción a MySQL para el almacenaje de los datos.

Con este último punto se pretende dar un vistazo menos teórico, ofreciendo una aplicación sencilla desde la cual podemos ver los datos recogidos en una página web.

4.1 Arquitectura

La descripción del sistema se hará a través de una diseminación de las diferentes fases por las cuales procede una transmisión. La arquitectura global del sistema se puede ver en la Figura 4.1.

La primera etapa es la sección denominada Sección LoRa. En ella los nodos recolectores de datos se comunican con el GW empleando el protocolo LoRa. Seguidamente se encuentra la Sección TTN, que engloba las etapas que involucran el intercambio sobre TCP/IP del GW con la red TTN. Por último, la Sección back-end/servidor se encuentra la base de datos, y la plataforma que proveerá la interfaz de usuario.

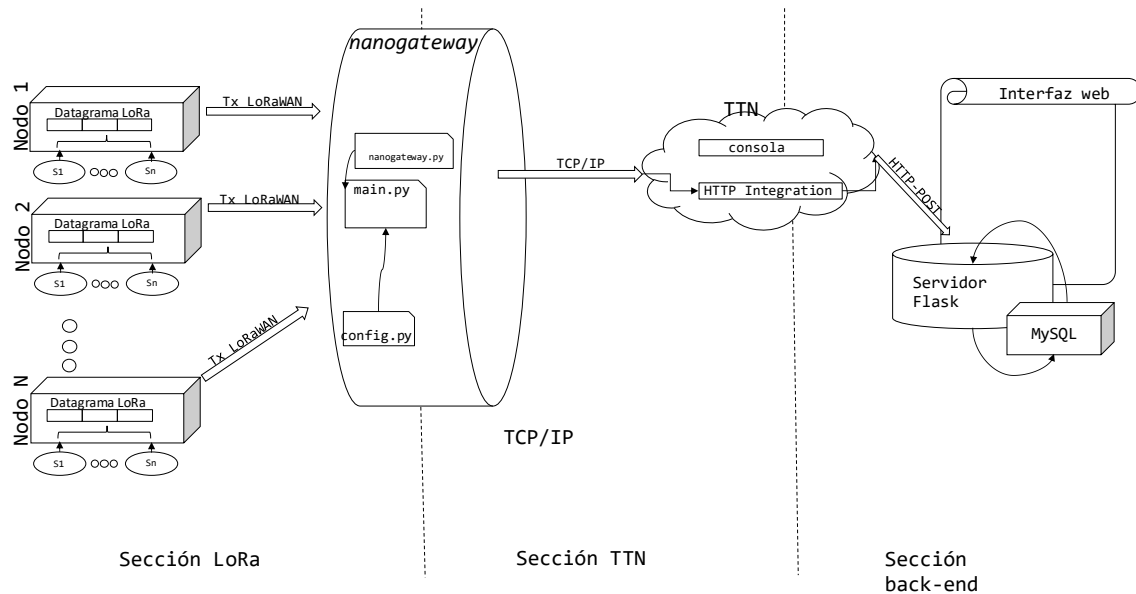


Figura 4.1 Esquema sistema LoRaWAN

4.2 Sección LoRa

La configuración hardware de los nodos transmisores incluye dos modelos de sensores:

- LoPy + Expansion Board: emula el comportamiento de un sensor que adquiere datos analógicos.
- LoPy+Pytrack: emula la operativa de un nodo de seguimiento basado en GPS.

Todos los datos recogidos por ambas opciones serán encapsulados para conformar el payload del mensaje.

El proceso de transmisión de los nodos se puede observar en la Figura 4.2.

A diferencia de la configuración empleada en la etapa de medidas y caracterización, se configuran los nodos para emplear el método de comunicación LoRaWAN.

```
lora = LoRa (mode=LoRa.LORAWAN)
```

A diferencia de las comunicaciones con LoRaRAW, implementar este protocolo conlleva la creación de un proceso de asociación definido en el estándar. Este proceso se da entre los nodos y TTN. Para el caso de nuestro despliegue, se ha elegido usar la opción de ABP para la autenticación. Ello supone que las claves no son negociadas, sino que se encuentran preestablecidas tanto en TTN como en los dispositivos. Así pues, se han de configurar tanto los nodos como TTN para incluir información correlada de direcciones de dispositivos aceptadas, y

las claves de red (Nwk_Skey) y aplicación (App_Skey). En la sección 4.3 se profundizará en cómo se fijan estos parámetros en la plataforma TTN.

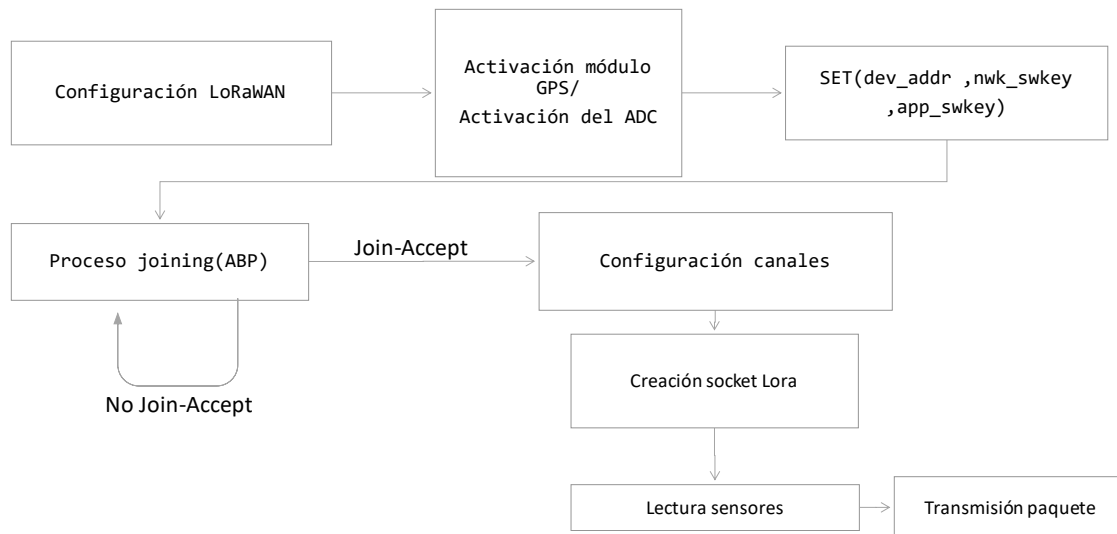


Figura 4.2 Esquema de transmisión LoRaWAN

4.2.1 Proceso de asociación

El próximo estado en una transmisión es el proceso de asociación con nuestro NS y AS. Para ello se envía un Join-Request hacia el GW, este redirecciona a través de TCP el mensaje hacia TTN, el cual se encargará de corroborar las credenciales del nodo. En caso de que el nodo este registrado en TTN y por tanto se tenga constancia de que puede unirse a la red, se responde con Join-Accept. Si no se recibe, el nodo quedará en bucle a la espera de unirse a una red LoRaWAN.

Una vez pasada esta etapa, se ha de gestionar los canales de comunicación a usar. Se opta por la creación de 3 canales habilitados para comunicarse con el GW. Este proceso se hace a través del siguiente código, en él se especifica el canal a añadir y la frecuencia de funcionamiento (frecuencia de funcionamiento de LoRa en Europa):

```
lora.add_channel(0, frequency=868100000,dr_min=0,dr_max=5)
```

Seguidamente y como etapa final antes del proceso de asociación, se crea un socket que vincule el GW con el nodo.

Una vez llegados a este punto, se tiene un nodo aceptado en la red, por ello se puede transmitir hacia TTN.

4.2.2 Recolección de información

Se procede a la lectura por parte del ADC de los pines en que hayamos conectado sensores, y se engloba toda la información para conformar el payload de nuestro mensaje. En este proyecto, se ha optado por el uso de un simple LDR que nos dé una estimación de la luz que se capta en la

localización del dispositivo. Además, se acompañará el mensaje con una lectura de la hora de envío, y un mensaje “Device X is here”.

4.2.3 Transmisión

Las transmisiones no son confirmadas, y a lo largo de las pruebas de funcionamiento se ha optado tanto por transmitir en intervalos prefijados, así como en caso excepcional haciendo uso de un botón y lectura de valor sobre este para hacer envíos cada vez que se detecte que es pulsado.

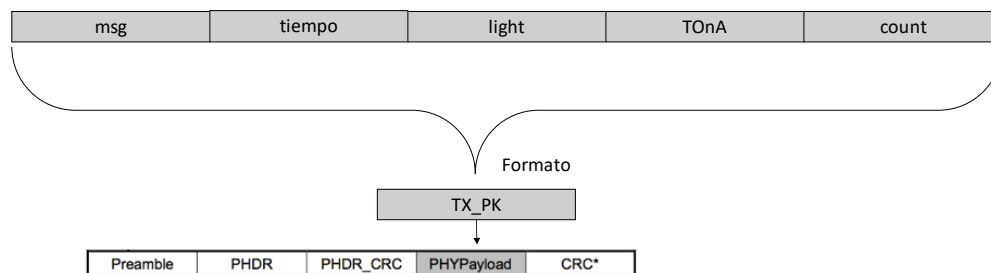


Figura 4.3 Formato paquete transmitido LoRaWAN

En la Figura 4.3 se muestra el formato de paquete. Las transmisiones se hacen con el mismo formato de paquete, este solo configurable a nivel de payload. Pues el resto del contenido es conformado según el estándar LoRaWAN. Véase las diversas cabeceras mencionadas en el capítulo 2.2.4

4.3 Sección TTN

La segunda etapa de comunicación por la que pasan los mensajes consiste en el redireccionamiento en el GW de los mensajes recibidos por la interfaz inalámbrica LoRa.

Las placas LoPy poseen, como se ha descrito en la sección 3.1, de un interfaz de comunicación WiFi. Se usará dicha funcionalidad para que todo mensaje que provenga de un dispositivo LoRa, se encapsule en un paquete TCP y se envíe a la plataforma habilitada por TTN.

Esta segunda sección abarca dos elementos de la red: el GW y TTN.

4.3.1 Gateway

El GW conforma el puente entre los entornos TCP/IP y LoRaWAN. Su funcionamiento se implementa mediante la ejecución de 3 programas simultáneamente:

- **Config.py**
Incluye la definición y almacenamiento de todas las variables constantes empleadas a lo largo del proceso de comunicación, tal como se refleja en la Tabla 9.

Tabla 9 Parámetros de configuración nanogateway

WIFI_MAC	ubinascii.hexlify(machine.unique_id()).upper()
GATEWAY_ID	WIFI_MAC [:6] + "FFFE" + WIFI_MAC [6:12]
SERVER	'router.eu.thethings.network'
PORT	1700
NTP	"pool.ntp.org"
NTP_PERIOD_S	3600
WIFI_SSID	'default'
WIFI_PASS	''
LORA_FREQUENCY	868100000
LORA_GW_DR	"SF7BW125"
LORA_NODE_DR	5

Se describen por tanto los campos requeridos para establecer la conexión WiFi, la sincronización del reloj empleando Network Time Protocol (NTP, Network Time Protocol), así como la dirección de TTN con su puerto de comunicaciones correspondiente. Además, se especifica el flujo de datos inicial a usar DR (DR, Data Rate), así como la dirección del GW y su frecuencia de funcionamiento.

- **Nanogateway.py**

En la Figura 4.4 se muestra el diagrama de flujo de la operativa del GW. Una vez es encendido siempre procede tal se describe a continuación:

- 1) Conexión a red WiFi especificada, en modo estación conectándose a un punto de acceso WiFi.
- 2) Una vez se establece la conexión se sincroniza el reloj a través del protocolo NTP.
- 3) Se establece una conexión IP al servidor TTN.

- 4) Se crea socket para comunicación sobre UDP entre el nanogateway y el servidor TTN.
- 5) Se configura a continuación las comunicaciones LoRaWAN que serán usadas para la transmisión entre nodos-nanogateway.
- 6) Solo cuando todos los pasos se hayan cumplido, el nanogateway se considerará activo y en funcionamiento.

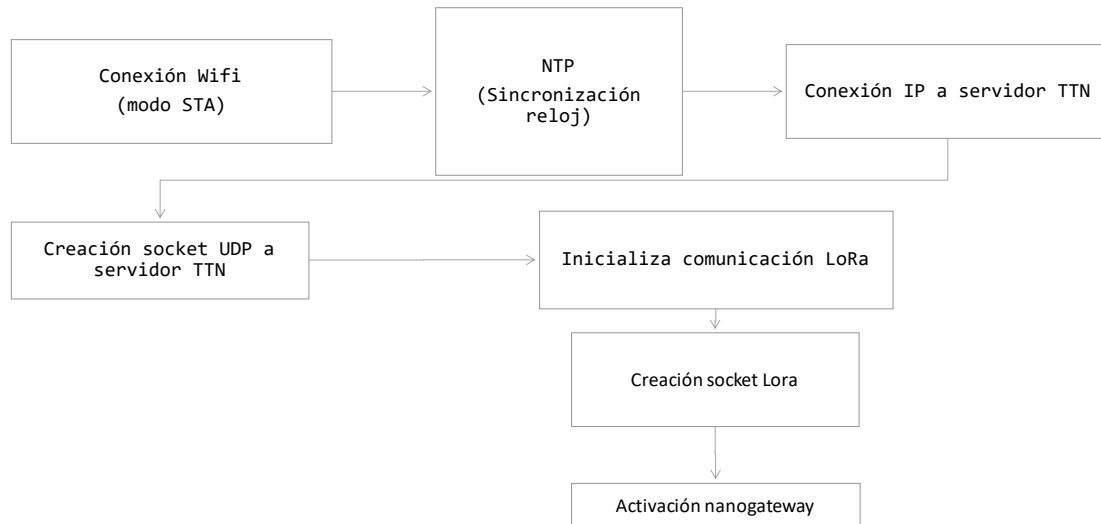


Figura 4.4 Activación del nanogateway

- **Main.py**

Sirve como nexo entre los parámetros preestablecidos e iniciar el programa **nanogateway.py**. Simplemente es el programa principal desde el que se ejecutan las funcionalidades requeridas, es decir la asignación de variables y la propia operativa del GW.

Cada vez que el nanogateway recibe un paquete por parte de un nodo, este inmediatamente añade a la información obtenida, las características de recepción LoRa de la transmisión a los metadatos a nivel MAC, de forma que todo sea accesible para su lectura desde el visualizador de TTN.

4.3.2 The Things Network

Los mensajes sobre LoRaWAN que lleguen al GW son redireccionados a TTN, la plataforma la cual conformará nuestro NS y AS. Como se ha mencionado en el apartado 2.3.1.1, en TTN se ha de hacer un paso previo para el establecimiento del sistema, el registro y generación de claves de nuestros dispositivos para que formen a ser parte de la plataforma.

Se tienen que hacer dos procesos de registro previo a la puesta en funcionamiento del sistema:

1. Registro de aplicación

Se vincula al registro del sistema o funcionalidad global que implementa el conjunto de nodos. A una aplicación se le vincularán a continuación el conjunto de dispositivos.

2. Registro GW

Una creación de la entidad que conformará nuestro GW, tal como se refleja en la Figura 4.5.

The screenshot shows the 'GATEWAY OVERVIEW' page in the TTN console. It displays the following information:

- Gateway ID:** eui-240ac4ffe0079e0
- Description:** Este es mi primer Gateway en el ciber mundo de las fantasías de LoRa
- Owner:** Nav14 (with a 'Transfer ownership' link)
- Status:** not connected
- Frequency Plan:** Europe 868MHz
- Router:** ttn-router-eu
- Gateway Key:** A field with a base64 encoded key and a 'base64' toggle.
- Last Seen:** 2 months ago
- Received Messages:** 747
- Transmitted Messages:** 10

Figura 4.5 Registro nanogateway

3. Registro nodo

Vinculado a una aplicación ya registrada, requiere establecer el nombre del dispositivo (DevID) y el método de activación que se empleará (ABP en nuestro caso). En la Figura 4.6 se incluye la información adicional a configurar. Tanto Dev_addr, Nwk_Skey y App_Skey son parámetros que el propio TTN nos ofrece de vuelta al completar el registro, los cuales han de configurarse en el dispositivo correspondiente.

The screenshot shows the 'DEVICE OVERVIEW' page in the TTN console. It displays the following information:

- Application ID:** nicofirstapp
- Device ID:** e01v2i3l
- Activation Method:** ABP
- Device EUI:** E011E012E013E014
- Application EUI:** 70B3D57ED0008B73
- Device Address:** 26011386
- Network Session Key:** A field with a base64 encoded key and a 'base64' toggle.
- App Session Key:** A field with a base64 encoded key and a 'base64' toggle.
- Status:** 27 days ago
- Frames up:** 221 (with a 'reset frame counters' link)
- Frames down:** 0

Figura 4.6 Registro nanogateway

Una vez se tienen todos los nodos registrados, y se ha implementado el sistema como se ha mencionado, se consigue un sistema LoRaWAN, pero como se ha optado por añadir más valor al sistema, queda la creación del back-end de la red.

4.4 Sección back-end/servidor

Una vez completados todos los registros, todas las funcionalidades de TTN son ofrecidas para su implementación en el sistema registrado. Una de las principales razones de la selección de TTN como NS y AS del sistema es debido a las funcionalidades que ofrece al usuario.

Como ya hemos descrito como su principal función es cumplir papeles en la red, pero TTN también ofrece herramientas al usuario, entre ellas una de las más recurridas en este proyecto es la consola para visualizar el tráfico. A través de ella podemos ver como se ve en la ilustración los mensajes que lleguen hasta este punto en la red.

No obstante, entre los objetivos del proyecto, se plantea la interacción de TTN con entornos externos que permitan el procesado y posterior presentación de la información al usuario final. En este sentido, se define un servicio web que recolectará los datos provenientes de TTN (previo registro) y almacenará y presentará de forma propia los datos al usuario.

Dicho sistema está conformado por un servidor propio, desarrollado en Flask [19], junto con una base de datos para el almacenamiento de los datos de la red construido en MySQL [20].

4.4.1 HTTPIntegration

Para el caso de este proyecto, se ha optado por usar la integración HTTPIntegration disponible en TTN. Esta permite redireccionar el tráfico que llega hasta TTN a través de primitivas HTTP a una URL. En la Figura 4.7 se muestra la consola de configuración TTN.

Para esta redirección del tráfico, se ha usado a su vez también el programa “localtunnel”[21], este habilita las comunicaciones a través del puerto 5000 del ordenador donde el servidor se establece, y proporciona una URL como puntero a dicho puerto, como dirección de destino.

INTEGRATION OVERVIEW

Process ID

getmygateway

Status

Running

Platform

HTTP Integration

(v2.6.0)

documentation

Author

The Things Industries B.V.

Description

Sends uplink data to an endpoint and receives downlink data over HTTP.

SETTINGS

Access Key

The access key used for downlink

default key

devices

messages

URL

The URL of the endpoint

https://fast-moth-77.localtunnel.me/processjson

Method

The HTTP method to use

POST

Figura 4.7 Configuración HTTP Integration

4.4.2 Servicio Web

Mediante Flask se desarrolla un servicio REST el cual se encargue en primer lugar y como función principal de captar las primitivas HTTP que llegan de TTN, y almacenar la información incluida en una base de datos. Adicionalmente el servicio proporcionará una interfaz para que los usuarios puedan acceder a la información almacenada. Servicio, mostrado en la Figura 4.8.

El diagrama ilustra el flujo de datos y las operaciones en el back-end de red LoRaWAN. A la izquierda, un recuadro gris etiquetado 'TX PK' envía una solicitud a un cilindro central etiquetado 'Flask'. La solicitud contiene 'URL: xxxxxxxx' y 'puerto: 5000'. El cilindro 'Flask' contiene una lista de endpoints: '/processjson', '/', '/name', '/search_counter' y '/search_dev_id', con 'MyApp.py' al pie. A la derecha, un cubo etiquetado 'Base de datos' contiene una tabla de 4x4. Se muestran tres operaciones: 'Insert()' desde '/processjson' hacia la base de datos, y dos operaciones 'select()' desde la base de datos hacia '/name' y '/search_dev_id'. En la parte inferior izquierda, un recuadro 'Interfaz web usuario' contiene los archivos 'Home.html', 'data.html', 'search.html' y 'layout.html', con una flecha que apunta hacia el cilindro 'Flask'.

Figura 4.8 Esquema de funcionamiento del back-end de red LoRaWAN

pág. 59

Procesado de información de TTN

Para poder captar los datos, hemos de habilitar un API REST (API, Application programming interface) que procese una petición POST. En nuestro caso el endpoint definido es `/processjson`.

4.4.2.1 En este endpoint se tiene disponible el procesamiento del mensaje JSON recibido, resultado de lo cual se tendrá un conjunto de datos compatibles con la estructura de la base de datos. Dicha base de datos se ha creado en función a las observaciones de los datos en el visualizador de TTN sobre MySQL con una tabla que englobe cada parámetro de los paquetes. En la Figura 4.9 se muestra la estructura de una de las tablas de dicha base de datos.

Info Columns Indexes Triggers Foreign keys Partitions Grants DDL						
Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
dev_id	varchar(20)		NO	utf8	utf8_general_ci	select,insert,update,
downlink_url	varchar(300)		YES	utf8	utf8_general_ci	select,insert,update,
counter	int(11)		YES			select,insert,update,
app_id	varchar(20)		YES	utf8	utf8_general_ci	select,insert,update,
payload_fields	varchar(400)		NO	utf8	utf8_general_ci	select,insert,update,
payload_raw	varchar(400)		YES	utf8	utf8_general_ci	select,insert,update,
hardware_serial	varchar(30)		YES	utf8	utf8_general_ci	select,insert,update,
port	int(11)		YES			select,insert,update,
metadata	varchar(500)		YES	utf8	utf8_general_ci	select,insert,update,

Figura 4.9 Tabla MySQL de aplicación REST

4.4.2.2 Aplicaciones implementadas en Flask

En torno a la base de datos, se configura la operativa GET del servicio REST implementando sobre Flask que habilite el acceso a los diversos datos almacenados.

Se incluyen varios endpoints:

- Página principal conteniendo una descripción junto con el acceso al resto de funciones. (`/`, Figura 4.10)
- Página a modo de visualizador, donde se exponen los datos recogidos, así como gráficas que se autorregulan (Figura 4.12) en función a la información que le llega. (`/names`, Figura 4.11)
- Página de búsquedas en función al ID del dispositivo (`/search_dev_id`) y en función al contador del mensaje (`/search_counter`).

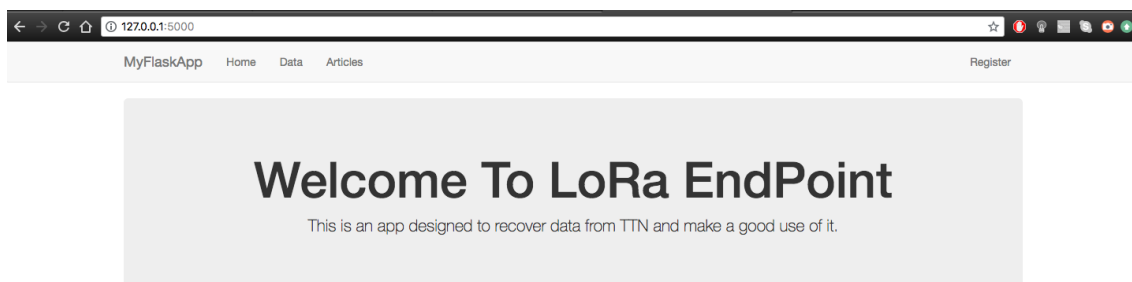


Figura 4.10 Interfaz usuario con servicio Web(/).

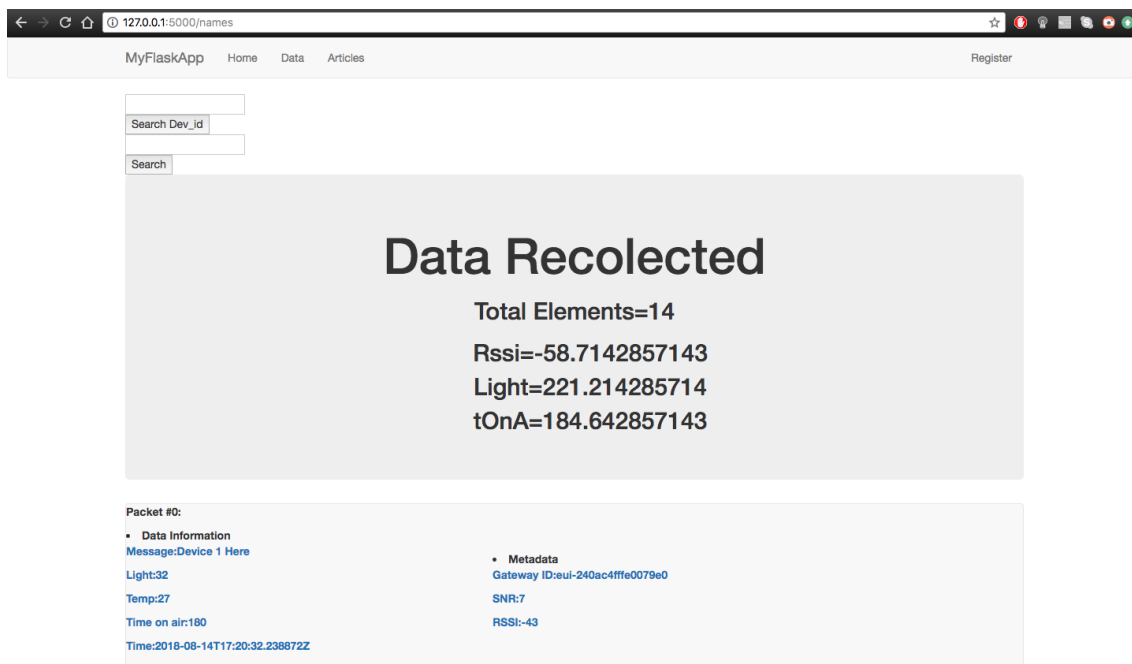


Figura 4.11 Interfaz usuario con datos(/names).

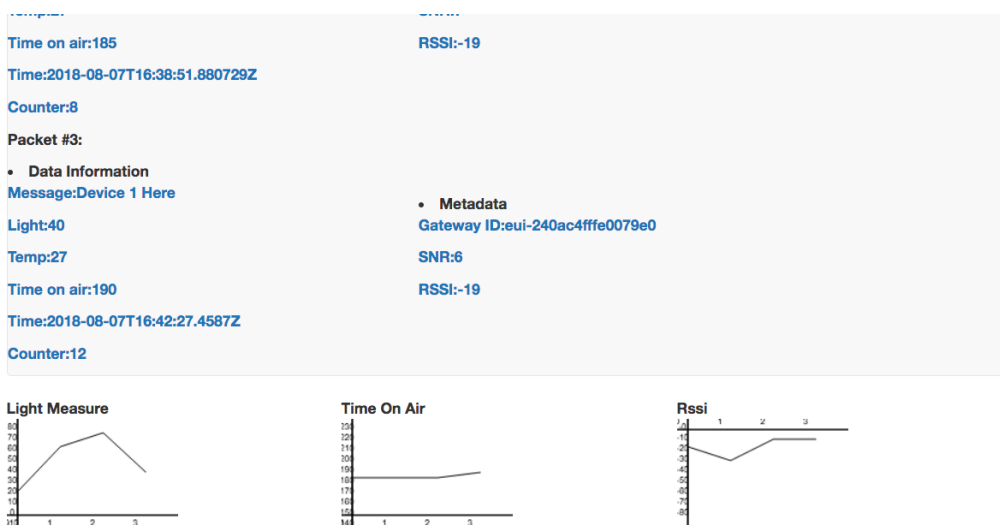


Figura 4.12 Gráficas autorreguladas en servicio Web

Así se consigue con la interacción entre todo ello que los datos obtenidos desde un punto inicial en la sección LoRa, atraviesen toda la sección hasta llegar a TTN. Y desde allí se redirijan al servicio REST para que estén disponibles para la visualización de un usuario a través de la interfaz que proporcionamos en el propio servicio. Con un despliegue a modo de página donde se permite la búsqueda de paquetes, y ver sus características.

5 Conclusiones y líneas futuras

Este capítulo conforma el capítulo de cierre del proyecto, buscando en este englobar todo aquello postulado en el proyecto. Una vez se ha evaluado diversos sistemas basados en LoRa, y desplegado un sistema LoRaWAN completo, basándonos en los procedimientos y resultados obtenidos se pretende aportar una serie de juicios.

Así conjunto de propuestas a futuro para su desarrollo sobre los postulados de este proyecto, que permitirían su expansión o el uso de su contenido para la creación nuevo material.

5.1 Conclusiones

En función a los sistemas LoRa y LoRaWAN que han sido desarrollados en los capítulos 3 y 4 se puede obtener conclusiones desde 2 perspectivas diferentes;

- Rendimiento y cobertura de redes LoRa
- Funcionalidad y viabilidad en despliegues LoRa y LoRaWAN

5.1.1 *Rendimiento y cobertura de redes LoRa*

Este apartado busca en base a los resultados obtenidos en las diversas campañas de medición poder concluir si una de las características ofrecidas por LoRa se verifica como son coberturas de hasta 15-20 Km en línea de visión directa.

Ante ello nos encontramos respecto a los resultados obtenidos en las campañas de medida de los capítulos 3.2.2 y 3.3.2 con un resultado que no se acerca a aquello que se oferta por parte de la tecnología.

A lo largo de las diversas pruebas de caracterización que se han hecho de los sistemas que solo recurrían a LoRa no se ha conseguido en ningún caso superar el kilómetro de cobertura. No por ello se concluye que esto sea una cobertura baja. Por un lado, la oferta teórica de LoRa matiza como no ha de haber ningún impedimento entre los extremos de comunicación, es decir, plantea un caso idílico.

En este proyecto se ha procedido a hacer despliegues en dos entornos diferentes, en los que sí se puede observar como en un primer entorno descrito en la Figura 3.9 se llega al horizonte con un SF de 7. Mientras que en el segundo entorno hecho en el apartado 3.3.2.2, en un entorno no tan beneficioso a la comunicación, se requiere de poner el SF a 12 para conseguir el mismo margen de cobertura.

Aún con todo ello, conseguir una cobertura de 1 Km en dos entornos diferentes es un área funcional para el despliegue de una red muy amplia, se puede considerar que la tecnología LoRa ofrece la posibilidad de despliegues que requieran abarcar un gran terreno. Parece que no es una

tecnología que presente un buen comportamiento en entorno semiurbanos o con obstáculos, como puede constatarse con los desvanecimientos ocurridos en las campañas de medición al llegar al borde de cobertura.

Durante el proceso de medición además se llegó a la conclusión de que las antenas a la hora de realizar la transmisión o recepción sobre LoRa requieren de un posicionamiento adecuado. Esto no ha sido descrito en un documento. Pero, si se procede a colocar el GW o los nodos con sus antenas pegadas muy cerca de una superficie o en posiciones horizontales ha conllevado en el desarrollo de este proyecto a un gran deterioro en sendos procesos de transmisión y recepción, llegando incluso a la pérdida de comunicación entre los dispositivos hasta corregir la posición de la antena.

5.1.2 Funcionalidad y viabilidad en despliegues LoRa y LoRaWAN

El principal objetivo de este proyecto se veía englobado en poder evaluar el funcionamiento de ambas tecnologías, para ello a lo largo del proyecto se ha recurrido tanto a despliegues como a caracterizaciones de estos, así como al montaje de un servicio web para visualizar un funcionamiento más práctico de estos sistemas.

En primer lugar, se quiere hablar respecto al funcionamiento de TTN. The Things Network es utilizada a lo largo de todo el capítulo 4. Para la utilización de esta plataforma, se requiere de hacer múltiples registros y configuraciones como han sido descritos. Todo el proceso para generar un despliegue de una red LoRaWAN basada en TTN se hace un proceso sencillo y muy tutelado. Tanto a través de Pycom como a través de la propia plataforma nos encontramos con procesos muy bien explicados y guiados, haciendo más accesible y recomendable el uso de esta.

El funcionamiento de la plataforma se corresponde a lo ofrecido, tanto la integración utilizada (HHTPIIntegration) como la conformación del NS y AS por parte de TTN funciona para ser usada como estructura para la creación de despliegues LoRaWAN.

Respecto a la red LoRaWAN como tal, una vez la red es conformada, tiene un modo de operación sobre ella muy simple, ya que podemos gestionarla a través de la plataforma y ver todo aquello que ocurre en ella. El funcionamiento para la transmisión de datos captados con diversos sensores como se explica en el capítulo 4.1 es correcto, pudiendo concluir que un despliegue LoRaWAN, es una opción viable para implementar sobre el concepto IoT.

Con la adquisición de diversos LoPy y sus correspondientes placas de expansión, podemos llegar a sensorizar áreas amplias, a un coste barato (aproximadamente 50€ incluyendo las antenas), siendo esto una alternativa al resto de tecnologías mencionadas en la introducción del proyecto.

En general, para un uso tanto profesional como para establecer una red más personal se concluye que el funcionamiento de un sistema basado en LoRaWAN es una opción factible y recomendable, la cual puede implementarse tanto recurriendo solo a las placas en el proyecto

utilizadas, sin requerir de costes adicionales, eligiendo una plataforma intermediaria para desplazar la información a una base de datos propia como se describe en apartado 4.

Ciertamente debido a que solo se han implementado redes con dispositivos de clase A, estos sistemas no son recomendables si se quiere implementar un gran número de nodos transmisores, el incremento gradual de transmisores en la red no es constatado en este proyecto, pero si es mencionado que conllevaría a la saturación de la red debido al modo de operación de los dispositivos clase A.

5.2 Líneas futuras

Ciertamente en este proyecto se hace un primer acercamiento a todas las posibilidades de esta tecnología, así como no se hace un estudio exhaustivo sobre ella. Entre las posibilidades que se considerar posibles y de interés a realizar, se encuentran:

- Realizar las implementaciones descritas en el proyecto con múltiples nodos, para poder llegar a hacer una caracterización más detallada del uso de los recursos de la red. Esto nos permitiría hacernos a la idea exacta del número máximo de dispositivos que pueden tenerse operando en una misma red modelar matemáticamente el sistema.
- Realizar un despliegue y evaluación de LoRa en un entorno completamente urbano y analizar las diferencias.
- Uso del método LoRaRAW para el desarrollo de un método de acceso a el canal similar a los dispositivos clase B de LoRaWAN (2.2.5), debido a que el método LoRaRAW nos permite la implementación libre de un protocolo que deseemos crear. Se tendría así la posibilidad de probar dispositivos pseudo-clase B que operen en LoRa a pesar de no encontrarse actualmente en el mercado.
- La creación de una plataforma propia sobre la que operar a través del código fuente de TTN, para poder tener como resultado una red LoRaWAN privada sin recurrir a manos de terceros para su desarrollo.
- Evaluar la tasa de fallos que se produce en los sistemas LoRa a través de la saturación de un canal radio con transmisiones continuas por parte de un nodo sobre el sistema del apartado 3.2

6 Bibliografía

- [1]. David Morrillas, “*IoT: Cuando la materia adquiere inteligencia*”. Artículo disponible en:
<http://www.synergicpartners.com/iot-cuando-la-materia-adquiere-inteligencia/>

- [2]. “LoRa Wireless Basics” By Hardy Schmidbauer, Director Wireless and IoT in Semtech. Disponible en:
https://www.electronicproducts.com/Digital_ICs/Communications_Interface/LoRa_wireless_basics.aspx

- [3]. “LoRa Spreading Factor & Data Rate” by Richard Wenner. Video disponible en:
<https://www.youtube.com/watch?v=B580NvdXtjs>

- [4]. Funcionamiento de la tecnología LoRa. Disponible en:
<https://www.alfaiot.com/index.php/es/2018/05/26/que-es-lora/>

- [5]. 330ohms, “¿Qué es SigFox y cómo funciona?”. Artículo disponible en:
<https://blog.330ohms.com/2017/05/11/que-es-sigfox-y-como-funciona/>

- [6]. LoRa Alliance Technical Committee., “*LoRaWAN 1.1 Specification*”, octubre 2017. Documento disponible en:
<https://loro-alliance.org/resource-hub/lorawantm-specification-v11>

- [7]. Información sobre la plataforma TTN y exposición de los servicios de TTN. Disponible en:
<https://www.thethingsnetwork.org/docs/>

- [8]. Restricciones y uso del duty_cycle en TTN. Disponible en:
<https://www.thethingsnetwork.org/docs/lorawan/duty-cycle.html>

- [9]. Información referente a la plataforma ResIOT. Disponible en:
<https://www.resiot.io/en/>

- [10]. Información referente a la plataforma TheThings.io. Disponible en:
<https://thethings.io/loro-iot-platform-dashboard/>

- [11]. Información sobre microchips RN para tecnología LoRa. Disponible en:
<https://github.com/TheThingsNetwork/ttn>

- [12]. Información del módulo SX1276RF1IAS. Disponible en:
<https://www.digikey.es/product-detail/es/semtech-corporation/SX1276RF1IAS/SX1276RF1IAS-ND/4490401>

- [13]. Información sobre la placa Arduino MKR WAN 1300. Disponible en:
<https://store.arduino.cc/mkr-wan-1300>

- [14]. Repositorio código de la plataforma TTN. Disponible en:
<https://github.com/TheThingsNetwork/ttn>
- [15]. Documentación relativa a la tecnología propiedad de Pycom. Disponible en:
<https://docs.pycom.io/>
- [16]. Lenguaje de programación Mycropython, finalidad y uso. Disponible en:
<https://micropython.org/>
- [17]. Repositorio de librerías propiedad de Pycom. Disponible en
<https://github.com/pycom/pycom-documentation>
- [18]. Funcionamiento y descripción del protocolo Aloha. Disponible en:
http://gordonbell.azurewebsites.net/computer_structures_principles_and_examples/csp0432.htm
- [19]. Implementación y funcionamiento de Flask. Disponible en:
<http://flask.pocoo.org/docs/1.0>
- [20]. Información sobre funcionamiento y despliegue de MySQL. Disponible en:
<https://www.mysql.com/>
- [21]. Repositorio sobre la directiva localtunnel. Disponible en:
<https://localtunnel.github.io/www/>

