Universidad Zaragoza
1542

Trabajo Fin de Grado

Diseño e Implementación de redes LoRa malladas

Design and Implementation of LoRa Mesh Networks

Autor

Ángel Torre Tolosana

Directores

Francisco José Martínez Domínguez

Julio Alberto Sangüesa Escorihuela

Escuela Universitaria Politécnica de Teruel
Universidad de Zaragoza
2022

Repositorio de la Universidad de Zaragoza – Zaguan http://zaguan.unizar.es

Diseño e Implementación de redes LoRa malladas

RESUMEN

El uso del Internet de las cosas (IoT) ha ido en aumento con los años y, gracias a la gran variedad de tecnologías de comunicación, se ha facilitado la obtención e intercambio de información en entornos de industria, salud, transporte, agricultura, ganadería, ciudades inteligentes, edificios y logística. Uno de los protocolos de transporte inalámbrico más básicos para la implementación de IoT es LPWAN (Low Power Wide Area Network), permitiendo la comunicación de un bajo volumen de datos entre grandes distancias y con un consumo mínimo. Existen diferentes tecnologías de radio LPWAN, una de ellas es el estándar de comunicación y transporte LoRa, que ha experimentado un crecimiento durante los últimos años para la implementación de soluciones IoT. LoRa es normalmente usada como capa física de LoRaWAN, un estándar abierto que permite la implantación de arquitecturas de red con topología en estrella.

A pesar del éxito de LoRaWAN en el mundo del IoT, puede experimentar pérdidas de información, ocasionadas por la distancia entre dispositivos que conforman la red, la topología del terreno y la creación de áreas de sombra por obstáculos. Además, el alcance de las comunicaciones que puede ofrecer LoRaWAN está limitado por su topología de red en estrella, donde los dispositivos envían la información al Gateway en un solo salto.

En este proyecto se propone la implementación de un sistema híbrido de recogida de datos basado en una red mallada unisalto, donde los nodos pueden comunicarse a través de LoRa para descubrir el camino más fiable al Gateway, enrutando la información a los nodos que tienen conexión con el servidor de red LoRaWAN. Esto permite transmitir los paquetes al servidor sin estar en una zona de cobertura, orientado su uso a entornos agrícolas donde los elementos de la red están estacionados de forma fija. El despliegue de la red LoRa mallada unisalto, ha requerido del diseño de protocolos para la gestión de comunicaciones entre los nodos que conforman la red, asegurando la eficacia y seguridad de los paquetes y, permitiendo a los nodos sin conexión, obtener los caminos disponibles situados a un salto de distancia que disponen de conexión al Gateway. También, se han diseñado las diferentes estructuras que conforman los paquetes enviados en la red, al igual que los métodos de empaquetado y desempaquetado, dando la posibilidad de enviar datos heterogéneos obtenidos por diferentes tipos de sensores.

La implementación de esta red LoRa mallada sobre las actuales redes LoraWAN, no solo mejora los problemas de cobertura presentados, sino que también aumenta el alcance efectivo para la obtención de datos fuera del rango del Gateway. Esta propuesta permitirá la aparición de nuevos servicios orientados al uso de IoT en entornos hostiles donde las redes LoRaWAN requieran de una mayor flexibilidad y cobertura.

Palabras clave

LPWAN, LoRa, LoRaWAN, Internet de las cosas (IoT), redes malladas

Contenido

Índice de figuras

1. INTRODUCCIÓN

El avance tecnológico en estos últimos años ha alzado también el sector del Internet de las cosas (IoT) y ha surgido una necesidad en la sociedad de estar en constante conexión con otras personas, objetos y entornos. El IoT ha sido una las tecnologías que se han aprovechado de esta necesidad, con implementaciones para mejorar, no solo la vida cotidiana de las personas, sino también distintos ambientes laborales, ya que las aplicaciones son casi infinitas.

Además, el reciente capítulo de la pandemia mundial, ocasionada por la COVID-19, ha propulsado todavía más el desarrollo y la implantación de nuevos sistemas para lograr entornos de trabajo más seguros y muchas empresas se han visto obligadas a la automatización de muchas operaciones. Algunos ejemplos son: la mejora de rastreo de posibles contactos infectados, uso de robots de limpieza, monitorización remota de pacientes, implementación de infraestructuras para trabajos a distancia o el uso de drones para obtener y difundir información.

Uno de los puntos fuertes del uso de IoT, es la posibilidad de obtener datos desde lugares remotos, donde el entorno dificulta el acceso o es casi imposible contar con una conexión a Internet. La versatilidad que nos brinda el poder realizar análisis en ambientes tan variados permite optimizar tareas de gestión y control y optimizar los costes. En este tipo de ambiente destaca el estándar LPWAN (Low Power Wide Area Network), uno de los protocolos de transporte inalámbrico que permite la comunicación mediante la tecnología de radio entre millones de dispositivos.

Una de las implementaciones del protocolo LPWAN es LoRa, que ha emergido en los últimos años como una opción popular y es usada como tecnología de radio que proporciona comunicación de largo alcance y una gran sensibilidad en el receptor. Además, puede ser usada de forma autónoma o en redes LoRaWAN como capa física. LoRaWAN es un protocolo abierto que implanta una arquitectura formada por diferentes nodos finales que envían información en un único salto a un servidor de red mediante las puertas de enlace (Gateway) conectadas. Esta tecnología es usada en una gran variedad de ámbitos, como edificios, ciudades inteligentes, industria y ganadería.

1.1. MOTIVACIÓN

El desarrollo industrial y la urbanización, así como el aumento del cambio climático, han impactado en la agricultura, ocupado una gran parte de tierra cultivable y creando crisis alimenticias. Al aplicar soluciones inteligentes sobre la agricultura, no solo se ayuda a mejorar la utilización de los recursos y aumentar el rendimiento de los cultivos, sino que también se optimizan los recursos humanos y se reducen el uso excesivo de químicos como pesticidas y fertilizantes. Se han implementado ya casos de uso como el monitoreo de consumo de agua y electricidad en tiempo real, clasificación de frutas y verduras, o medidores de temperatura y humedad, creando granjas inteligentes cuyo objetivo es maximizar el rendimiento de los cultivos en el menor espacio posible [1].

Debido a la gran importancia que tiene este sector, este proyecto se ha enfocado el uso de las tecnologías LoRa y LoRaWAN para crear un sistema de recolección de datos sobre sus tierras y cultivos, impulsando más el uso del IoT en este ámbito y permitiendo el desarrollo tecnológico de estas áreas.

LoRaWAN ha demostrado tener éxito en los ámbitos del IoT, sin embargo, entornos como el que se ha descrito se beneficiarían más si la topología de estrella fuera más flexible. La solución aportada va más allá de la implantación de una tradicional arquitectura LoRaWAN para la monitorización de datos. Se pretende realizar un estudio que combine la tecnología LoRaWAN, con una topología mallada basada en comunicaciones LoRa de un único salto para crear una solución que aporte una mayor flexibilidad, alcance y recuperación de errores que la tradicional LoRaWAN.

Implementar una red mallada LoRa de más de un salto aportaría una mayor tolerancia a errores, enrutando los paquetes por diferentes caminos en función de la disponibilidad de los dispositivos, además de permitir un mayor alcance en la obtención de datos. Sin embargo, se ha decidido limitar las comunicaciones de la red mallada LoRa a un único salto, ya que, contemplar una solución de más de un salto ocasionaría con cada envío una mayor congestión de tráfico en el espectro de radiofrecuencia al tratar de enrutar los paquetes, así como un aumento en el consumo de energía de los nodos, comprometiendo su autonomía.

1.2. OBJETIVOS

El objetivo principal, es el diseño e implementación de una red LoRa mallada que dote de mas flexibilidad al estándar LoRaWAN, dando como resultado un sistema capaz de obtener información de ayuda en entornos agrícolas.

Esta implementación debe cumplir los siguientes objetivos:

- Correcta combinación y uso de las comunicaciones P2P LoRa en la red mallada y la arquitectura LoRaWAN.
- Diseño de un protocolo en la red LoRa mallada que permita a los nodos situados fuera del Gateway descubrir el camino más fiable hacia los nodos con conexión.
- Diseño de un protocolo que gestione las comunicaciones en cada nodo de la red LoRa mallada.
- Sistema confiable que deje constancia de la recepción de la información.
- Optimización de la energía consumida por los dispositivos.
- Verificación de la solución final mediante la realización de pruebas a diferentes distancias entre dispositivos para validar su comportamiento.

1.3. FASES DE PROYECTO

Para la realización correcta de este proyecto se establecen tres fases claras a llevar a cabo, para que la propuesta final sea la versión más pulida posible y se logre cumplir con todos los requisitos y desafíos que se presentan.

1.3.1. FASE DE INVESTIGACIÓN

Como cada vez que se empieza un proyecto en un tema desconocido, lo primero es la investigación acerca del tema en cuestión para ser conscientes de las limitaciones y capacidades que disponemos. Es necesario realizar un proceso de investigación sobre la tecnología LoRa y sobre su uso en el estándar LoRaWAN. Debe entenderse cómo funcionan las comunicaciones entre dispositivos LoRa, cuáles son sus limitaciones y que funcionalidades se deben añadir para cumplir los objetivos del proyecto.

También es necesario el estudio de las comunicaciones de los nodos finales con los Gateway y su envío de datos al servidor en una arquitectura LoRaWAN.

Finalmente, la investigación acerca del hardware compatible y adecuado que se necesita para la implementación y correcto funcionamiento de las comunicaciones LoRa.

1.3.2. FASE DE DESARROLLO E IMPLEMENTACIÓN

Tras la fase de investigación, y siendo conscientes de las distintas opciones que podemos realizar, se debe realizar la implementación real del sistema siguiendo la filosofía y capacidades de la tecnología LoRa.

Se realiza la programación del comportamiento que deben tener los dispositivos que componen el sistema, con respecto a otros nodos, en función de su posición con el Gateway, cómo se envían y reciben los datos, funciones que hagan un sistema fiable, visualización final de los datos en el servidor, y la correcta gestión de las librerías utilizadas.

Al final de esta fase se debe tener un prototipo sólido sobre el que realizar pruebas para la detección de posibles errores y mejoras.

1.3.3. FASE DE VALIDACIÓN

Durante esta fase se debe poner a prueba el sistema implementado y comprobar todos los escenarios posibles que pueden darse en una implementación real con la finalidad de encontrar fallos y mejoras.

Se analiza el sistema y su comportamiento ante las posibles adversidades, se asegura que el sistema responde de la manera esperada, se observa si hay pérdidas de información, la capacidad para recuperarse de errores y la flexibilidad del sistema para encontrar caminos que lleven la información a su destino.

1.4. ESTRUCTURA DE LA MEMORIA

En la sección 2, se expone un conjunto de implementaciones físicas que se han llevado a cabo mediante la tecnología LoRa y LoRaWAN para el uso del IoT en diferentes ámbitos. La intención es dar contexto sobre la utilización de la tecnología que se implementa en este proyecto, para tener un enfoque de la situación actual.

En la sección 3, se describen en detalle las tecnologías utilizadas de comunicación, tanto los protocolos de la red mallada y LoRaWAN, como los componentes hardware y software elegidos para la implementación física.

A lo largo de la sección 4, se exponen los problemas que se han enfrentado durante el desarrollo del proyecto y las soluciones aplicadas. Se detalla el funcionamiento del sistema general y las funciones que lo componen implementadas en el código.

En la sección 5, se muestran los resultados de la implementación física sometida a diferentes tipos de pruebas de distancia entre dispositivos.

Finalmente, se presenta en la sección 6 las conclusiones obtenidas una vez el proyecto ha sido desarrollado.

Como información anexada, podemos encontrar el proceso de configuración tanto hardware como software de las tecnologías utilizadas.

2. ESTADO DEL ARTE

Se presentan en este apartado una variedad de soluciones reales que hacen uso de la tecnología LoRa y la arquitectura LoRaWAN en diferentes ámbitos. En la Figura 1, se pueden observar los diferentes campos que implementan soluciones basadas en comunicaciones LoRa para proyectos IoT. En este apartado se aportan diferentes implementaciones reales que han llevado a cabo diferentes empresas en algunos de estos ámbitos.

Figura 1. Ámbitos de aplicación de la tecnología LoRa [2]

2.1. INDUSTRIA INTELIGENTE

La industria puede beneficiarse de la tecnología LoRa y el despliegue de sensores conectados a IoT para la realización de múltiples funciones de monitoreo activo. Cada vez es más común el uso de estas tecnologías en el sector de la industria y temas de logística.

2.1.1. MONITORIZACIÓN DE INSTALACIONES Y ACTIVOS

Cisco desarrolló soluciones de sensores IoT para mejorar la visibilidad en entornos de TI y tecnología operacional. Esta solución, llamada Cisco Industrial Asset Vision, permite la monitorización y administración remota de activos en las instalaciones mediante el protocolo LoRaWAN [3]. Es útil para el seguimiento de activos en interiores y exteriores, control de temperatura y humedad, detección de fugas de aguas, condiciones de iluminación, detección de vibraciones y movimiento de equipo para prevenir robos y mejorar la logística. Esta implementación permite a los clientes de Cisco beneficiarse de la conectividad prestada para monitorizar sus activos de forma remota desde la nube**.**

2.1.2. GESTIÓN DE GAS PETRÓLEO

AIUT, un proveedor de hardware y software especializado en soluciones IoT, ha llevado a cabo una solución que mide el volumen de gas licuado de petróleo en los tanques, gracias a la implementación de dispositivos LoRa y el protocolo LoRaWAN [4]. Se realiza un seguimiento remoto y automatizado del consumo de gas, y mediante esta monitorización, se mejoran los

procesos de gestión de suministros y reserva de las distribuidoras, lo que les permite reducir los costes operativos y mejorar los flujos de trabajo. Esta nueva plataforma de AIUT es la solución que buscan las estaciones de combustible y proveedores, ya que necesitan estimar los niveles de tanques y optimizar los programas de recarga.

2.2. AGRICULTURA Y GANADERÍA INTELIGENTE

En el sector agrícola y ganadero, con la aparición de nuevas tecnologías, han aparecido también términos como "Smart Farming" o Granjas Inteligentes. La metodología de estas granjas es aplicar la tecnología en sus trabajos de forma que les facilite y optimice las tareas. El monitoreo en granjas es un gran cambio que puede ser implementado en una gran variedad de ámbitos rurales gracias a la larga duración, alcance y variedad de dispositivos sensores. LoRa es una de las fuertes apuestas en tecnología para la recolección de datos en la industria agrícola y animal.

2.2.1. MONITORIZACIÓN EN TIEMPO REAL DE PARÁMETROS AMBIENTALES

La empresa PrognostiX diseñó una solución llamada Poultry Sense IoT para la monitorización en tiempo real de parámetros ambientales y de salud sobre las aves en las granjas avícolas para mejorar el rendimiento y bienestar [5]. Se combinan una serie de sensores que monitorean la temperatura, humedad, consumo de agua y peso de las aves. Estos datos captados por el sistema son analizados por especialistas veterinarios, ingenieros y diferentes analistas para obtener resultados y ayudar a la toma de decisiones. Esta infraestructura usa la conectividad LoRaWAN para proporcionar una comunicación bidireccional a bajo costo, largo alcance y escalable.

2.2.2. AUMENTO DE TASAS DE FERTILIDAD ANIMAL

Webee es una empresa californiana que ha implementado un sistema de monitorización que utiliza nodos sensores LoRa sobre una red LoRAWAN para monitorear incubadoras de huevos con la finalidad de mejorar la tasa de eclosión de huevos [6]. Capturan información en tiempo real sobre temperatura, humedad, ventilación y condiciones ambientales.

El uso de LoRaWAN hace que la infraestructura sea muy sencilla, con una puerta de enlace que da cobertura a un área muy extensa y sensores de bajo costo. Este sistema previene con avisos como SMS cuando la temperatura salga de los limites óptimos.

2.2.3. MONITORIZACIÓN DE CULTIVOS

MOKOLoRa implementa una solución que monitorea la humedad y temperatura en entorno de plantación gracias a la tecnología LoRaWAN y mediante el uso de sus sensores inteligentes [7]. Disponen de un sistema automatizado que activa una alarma siempre que aparezcan cambios no deseados en la temperatura y humedad que activarán unos sistemas de riego, humidificación y control de temperatura de forma automática cuando sea requiera. En la Figura 2 se puede observar como la información recibida por los sensores es enviada al servidor LoRaWAN a través de la puerta de enlace. Los datos son mostrados en una aplicación externa, en la cual se procesa la información y automatiza los procesos que envían la señal de activación a los sistemas de humedad y temperatura.

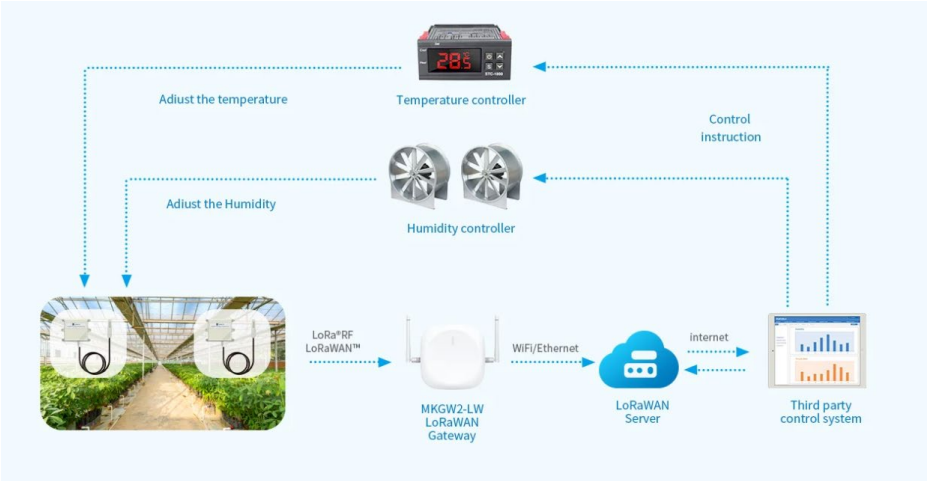

Figura 2. Sistema de monitorización automatizado MOKOLora [7]

2.3. CIUDADES INTELIGENTES

La tecnología LoRa y las redes de área amplia y baja potencia o basadas en el protocolo LoRaWAN, aportan una infraestructura de control con capacidad para captar y analizar datos de miles de sensores conectados para generar resultados y decisiones sobre los servicios. La tecnología de ciudades inteligentes está logrando un cambio en la forma de interacción entre ciudades – gobiernos – ciudadanos y hay un montón de aplicaciones como la gestión de estacionamientos, gestión de agua y control de gases**.**

2.3.1. GESTIÓN DE APARCAMIENTOS

Con motivo de ahorrar tiempo a la hora de buscar un aparcamiento libre, PNI Sensor Corporation propone una solución que ayude a la localización de sitios vacantes de estacionamiento en tiempo real y de forma remota [8]. Esta implementación necesita de un sensor de estacionamiento inteligente de alta precisión, el cual es montado en el suelo o superficie. Estos sensores se sitúan sobre zonas de estacionamiento y mediante la tecnología LoRa se comunican con una puerta de enlace que les brinda acceso al IoT. Esta información se comparte con las aplicaciones de terceros, que notificarán a los conductores sobre estacionamientos libres a través de su smartphone.

2.3.2. ADMINISTRACIÓN INTELIGENTE DE ENERGÍA

La empresa KingTingTech (YoSmart) desarrolló una solución para hogares inteligentes basada en LoRa [9]. Dispone de un centro de control que se comunica con los sensores inalámbricos integrados para transmitir datos a unos termostatos inteligentes y controladores de iluminación. Estos termostatos y controladores de luz también pueden ejecutar tareas automatizadas basadas en la estación del año y horarios para el ahorro de energía. La tecnología LoRa permite que un módulo de nodo final del sensor dure más de 10 años sin remplazo de baterías.

2.3.3. MEJORA DE REDES INALÁMBRICAS EN CIUDADES INTELIGENTES

Un protocolo llamado JMAC [10] pretende solucionar los problemas de conectividad donde la visibilidad directa y cobertura es reducida. Este protocolo de salto múltiple utiliza la tecnología LoRa para realizar comunicaciones de bajo consumo y amplia distancia, diseñado para mejorar redes inalámbricas de largo alcance y permitiendo la monitorización de ciudades inteligentes o

industriales. JMAC aprovecha el comportamiento de malla para mejorar la cobertura y aumentar el número de dispositivos en la red. Esta solución fue analizada bajo diferentes condiciones en el simulador OMNet++ hasta obtener su validación.

2.3.4. MONITORIZACIÓN DE UN ÁREA GRANDE MEDIANTE RED MALLADA LORA

Los estudios realizados en la Universidad Nacional Chung-Cheng (Taiwan), observaron que en zonas urbanas LoRa se requiere de la implementación de una gran cantidad de Gateways para la recepción de datos de los dispositivos. Para evitar añadir Gateways adicionales, se hizo un despliegue a lo largo del campus de 19 nodos de malla LoRa con una solución multisalto [11]. El Gateway central coordina la recogida de información, recopilando datos a intervalos de 1 minuto, permitiéndoles aumentar el rango de comunicación y la tasa de entrega de datos.

2.4. NATURALEZA INTELIGENTE

Mediante la tecnología LoRa se pueden facilitar el envío de indicadores ambientales que aporten información para la realización de análisis que ayuden a la prevención y detección de problemas, antes de que sea demasiado tarde.

2.4.1. PRESERVACIÓN DE BOSQUES URBANOS

ICT International, proveedor de soluciones IoT orientada a aplicaciones ambientales ha implementado un sistema para mejorar la gestión forestal urbana y la monitorización de los niveles de carbono, mediante el uso de dispositivos LoRa y LoRaWAN [12]. Se monitoriza la salud del bosque en tiempo real a través de LoRaWAN mediante mediciones del uso del agua de las plantas (flujo de savia), humedad del suelo y déficit de presión de vapor. Además, este sistema también es capaz de identificar periodos de baja humedad en el suelo y alto estés hídrico en las plantas. Esta solución contribuye a la conservación de parques y bosques urbanos manteniéndolos saludables y ayudando al ecosistema.

2.4.2. MONITORIZACIÓN DE INUNDACIONES

Una empresa encargada en la implementación de tecnologías ambientales llamada Green Stream desarrolló una solución con dispositivos inalámbricos y comunicaciones por radio frecuencia LoRa para construir un sistema autónomo en monitoreo de inundaciones que permitiese la construcción de comunidades más seguras en áreas costeras [13]. Cada sensor es desplegado a lo largo de la orilla, pantanos u otros terrenos accidentados y dispone de un pequeño panel solar del que recoge energía. Estos datos obtenidos son enviados en la red LoRaWAN y gestionados para su estudio y búsqueda de resultados.

2.4.3. DETECCIÓN DE INCENDIOS FORESTALES CON REDES MALLADAS LORA

Para prevenir y facilitar el manejo de incendios forestales, en Indonesia se realizó un despliegue de nodos sensores en medio del bosque, capaces de medir la temperatura, humedad y concentraciones de gas [14]. Estos nodos utilizan la red mallada LoRa para comunicarse a lo largo del bosque y detectar la ubicación del incendio, pudiendo enviar la información al Gateway a una distancia de 500 metros. Para este despliegue utilizaron 4 nodos colocados a 100 metros cada uno, logrando evitar las colisiones entre ellos y consiguiendo un nivel RSSI de -136dBm en el Gateway.

2.4.4. SEGUIMIENTO DE ANIMALES MEDIANTE REDES MALLADAS LORA

Para conservar especies animales en peligro de extinción, se propone una solución basada en redes malladas LoRa [15], donde a través de nodos en forma de collar o etiqueta, se puede

rastrear la ubicación de los animales y otros parámetros de interés relacionados con la actividad de los animales. Esta solución de red mallada LoRa, permite la comunicación entre grandes distancias en la sábana, el desierto y los bosques, una opción mucho más barata que el seguimiento por satélite o el despliegue de redes celulares remotas.

3. TECNOLOGÍAS EMPLEADAS

Como ya se ha comentado, las tecnologías de comunicación LPWAN han obtenido una gran importancia en el sector de IoT, gracias al soporte de grandes distancias de transmisión y unos requisitos de ancho de banda bajos. Sin embargo, hay que ser conscientes de la existencia de otras alternativas para las comunicaciones inalámbricas en el ámbito del IoT. Las principales alternativas son: la tecnología Wi-Fi, Bluetooth, BLE, ZigBee, LTE-M, el protocolo NB-IoT y las redes 5g. Estas tecnologías pueden dividirse en función de su alcance, corto, medio y largo. (Ver Figura 3).

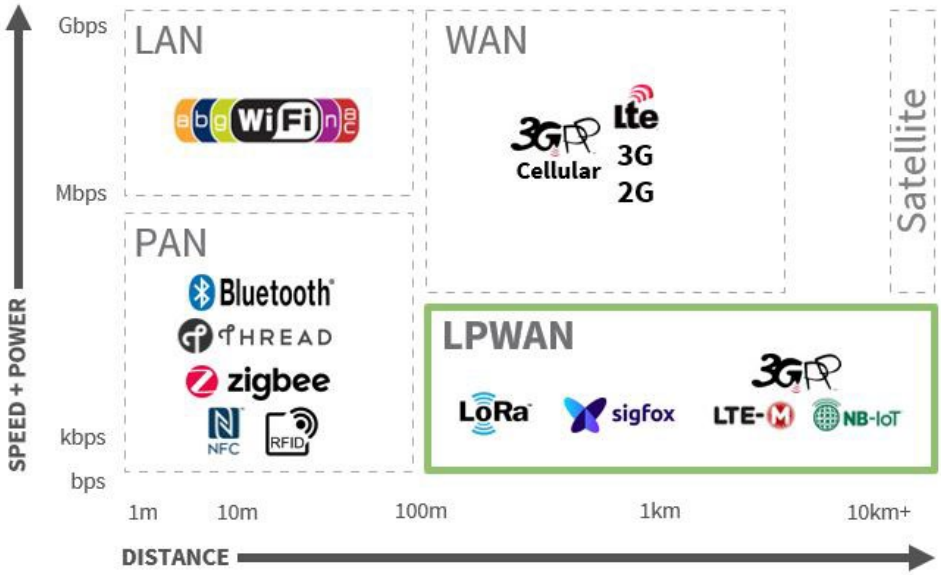

Figura 3. Comparación tecnologías inalámbricas [16]

3.1. LORA

LoRa es una tecnología de comunicación inalámbrica que utiliza una modulación de radio frecuencias desarrollada y patentada por la compañía SemTech [17]. Esta tecnología de modulación está basada en Chirp Spread Spectrum (CSS). Los llamados pulsos de chirp se envían como símbolos que aumentan o disminuyen la frecuencia de LoRa continuamente con el tiempo. Los datos son transmitidos por la secuencia de estos barridos lineales llamados chirps, como se puede ver en la Figura 4. Gracias a esta modulación, se ofrece una alta tolerancia al ruido e interferencias a lo largo de una cobertura de varios kilómetros.

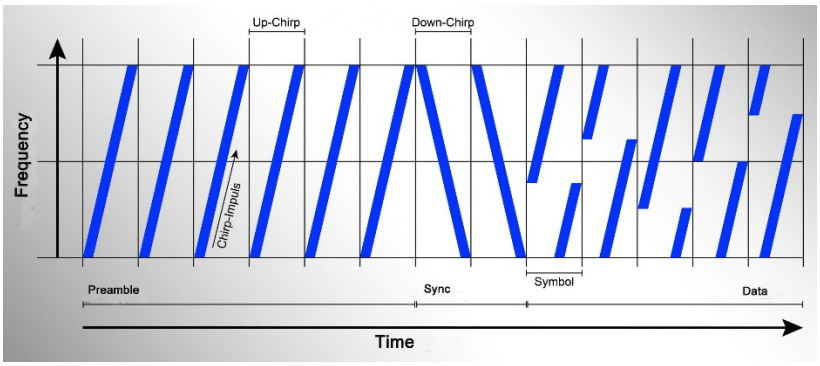

Figura 4. Barridos de frecuencia LoRa [18]

Funciona en las bandas de frecuencia 433MHz, 868MHz y 915MHz asignadas en los distintos países (ver Figura 5). En España se opera en banda no licenciada del radioespectro, lo que implica poder transmitir en la frecuencia 868 de manera gratuita. A nivel europeo, el European Telecommunications Standards Institute (ETSI) [19] define las siguientes limitaciones:

- Bandas de frecuencia: 868-869 MHz
- 10 canales: 8 con data rates de entre 250bps y 5.5 kbps, 1 de 11kbps y 1 FSK de 50kbps
- Potencia de salida máxima: +14 dBM.
- Duty Cycle de entre 0,1% y 1%, lo que limita el uso del espectro electromagnético a un tiempo de comunicación diario de 15 minutos máximo.
- En redes privadas no hay limitación de tiempo de permanencia en el canal

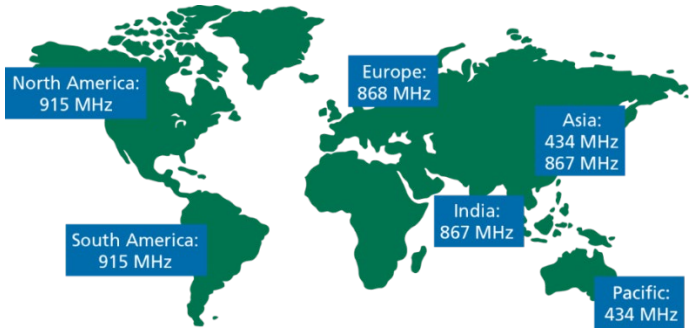

Figura 5. Frecuencias LoRa ISM [20]

El consumo energético en dispositivos LoRa es optimizado de manera eficiente gracias a los factores de dispersión ortogonales, que permiten adaptar a cada nodo las características de transmisión según la situación. Los factores de dispersión o SF (Spreading Factors) se clasifican en seis distintos (desde SF7 hasta SF12) y controlan la tasa de chirp, es decir, la velocidad de transmisión de datos, influyendo también en el tiempo en el aire, la batería y la sensibilidad del receptor. Los factores de dispersión más bajos reducen el rango de las transmisiones, al reducir la ganancia de procesamiento y al aumentar la tasa de bits. También son utilizados para controlar la congestión en la red ya que, al ser ortogonales, las señales moduladas con diferentes factores de dispersión y transmitidas en el mismo canal al mismo tiempo no van a interferir.

LoRa Spreading Factors (125kHz bw)

Spreading Factor	Chips/symbol	SNR limit	Time-on-air (10 byte packet)	Bitrate
7	128	-7.5	56 ms	5469 bps
8	256	-10	103 ms	3125 bps
9	512	-12.5	205 ms	1758 bps
10	1024	-15	371 ms	977 bps
11	2048	-17.5	741 ms	537 bps
12	4096	-20	1483 ms	293 bps

Figura 6. Impacto de niveles SF [21]

La tecnología LoRa proporciona una capa física que permite las comunicaciones punto a punto y punto a multipunto entre dispositivos compatibles. Lo que no permite, es el envío direccional de las transmisiones hacia un destino concreto. Todos los envíos de la información se realizan en modo broadcast. Este dato es limitador a la hora de hacer una gran red mallada de nodos ya que, crearía una gran congestión en la red.

SemTech diseña y ensambla los chips que se implementan en estos dispositivos, lo que da una gran variedad de hardware compatible con la tecnología LoRa, permitiendo que aplicaciones inteligentes de IoT resuelvan una amplia variedad de necesidades como la administración de energía, recursos naturales, control de contaminación, eficiencia en infraestructuras y prevención de desastres. Ya son muchos los casos que implementan esta tecnología para lograr ciudades y hogares inteligentes, así como en el sector de agricultura y ganadería, que han logrado mejorar el rendimiento y calidad de su trabajo. LoRa se ha convertido en una opción poderosa a la hora de desarrollar proyectos para IoT, sobre todo en lugares remotos o poco accesibles, debido a sus bajo consumo energético y largo alcance.

3.2. LORAWAN

LoRaWAN es un estándar de red de área amplia (LPWAN) de baja potencia administrado por LoRa Alliance que define una arquitectura y un protocolo de comunicación basado en la capa física LoRa. Como ya se ha comentado, cumple con los requisitos para su uso en el IoT, con una comunicación bidireccional, seguridad de extremo a extremo y servicios de geolocalización.

A diferencia de las demás LPWAN, LoRaWAN permite el despliegue de redes propias autogestionadas. Al ser una red autogestionada, solo se restringe a los límites del ETSI vistos anteriormente, que son menos estrictos que los que presenta un operador privado. Este hecho abre un gran abanico de posibilidades para realizar iniciativas IoT. Permite montar redes seguras en cualquier lugar, con distintos métodos para autenticar dispositivos, encriptación de datos y un sistema fiable en la entrega de datos, además de unos costes de mantenimiento menores.

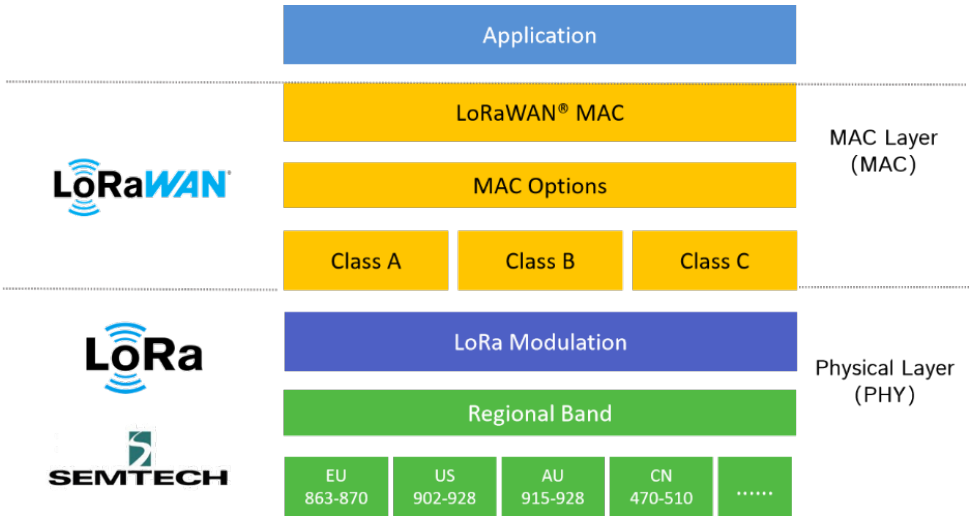

Figura 7. Capas LoRaWAN [22]

3.2.1. ARQUITECTURA

LoRaWAN funciona con una topología en estrella, donde la estructura de comunicaciones está constituida por (Ver Figura 8):

- Nodos finales: Puede ser un sensor, un actuador o ambos. Están conectados inalámbricamente a la red LoRaWAN e intercambian información con el servidor, a través de las puertas de enlace.
- Servidor: Administra las puertas de enlace, dispositivos finales y aplicaciones. Es capaz de comunicarse con los nodos.
- Puertas de enlace (Gateways): Están registradas en el servidor y tendrán asociados los nodos finales de los cuales recibirán información, siempre que estén dentro de su área de cobertura, y la reenviarán al servidor LoRaWAN. Se comunicán con el servidor de red mediante (3G/4G/5G), WiFi, Ethernet, fibra óptica o enlaces de radio de 2,4 GHz.
- Aplicaciones asociadas a la red LoRaWAN: Dentro de una misma red LoRaWAN pueden haber asociadas distintas aplicaciones, para gestionar los datos recibidos en el servidor.

Al estar basada en una topología en estrella, los nodos finales se comunican con el Servidor en un único salto. Sin embargo, estas comunicaciones no pueden suceder al mismo tiempo (enlace half-duplex). La dirección de las comunicaciones se distingue por mensajes Uplink (dispositivo emisor y servidor receptor) y Downlink (dispositivo receptor y servidor emisor). Uno de los problemas que surge en las redes LoRaWAN es la aparición de áreas de sombra, donde la señal del nodo final no alcanza al Gateway, debido a obstáculos o a la topología del terreno.

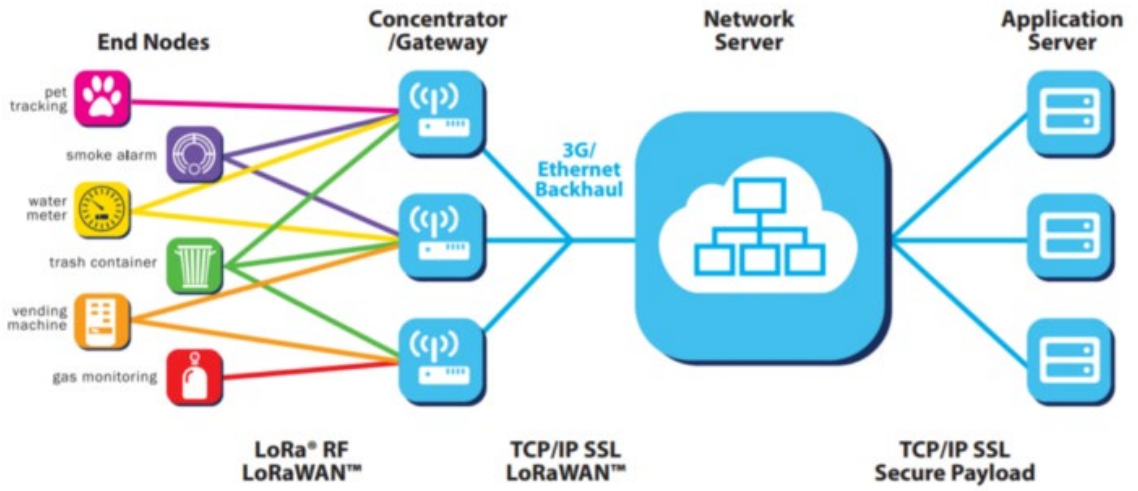

Figura 8. Arquitectura LoRaWAN [23]

3.2.2. CLASES DE NODOS FINALES

Todos los nodos finales de una red LoRaWAN deberán mantener una ventana de recepción abierta después de realizar una transmisión, en la que se espera recibir una confirmación ACK de este envío por parte del servidor. Se distinguen 3 clases de nodos finales, en función de su capacidad para recibir mensajes tipo Downlink del servidor:

- Clase A: Comunicación bidireccional. Después de cada transmisión Uplink, se habilitan dos ventanas cortas para la recepción del Downlink.
- Clase B: Comunicación bidireccional. Cuenta con ventanas de recepción adicionales, que están sincronizadas de forma temporal con el Gatway. Para poder abrir esta ventana de recepción, el dispositivo final debe recibir un Beacon desde el Gateway para que el servidor pueda saber cuándo está escuchando.
- Clase C: Tiene continuamente una ventana abierta de recepción que únicamente se cierra durante la transmisión. Esta modalidad ofrece una menor latencia en la comunicación servidor-nodo final, pero conlleva un gran consumo de potencia en comparación con las otras clases.

La elección de una de estas clases depende de la orientación que requiera el proyecto. Es una de las virtudes que posee LoRaWAN en cuanto a adaptación en distintos entornos. Por ejemplo, los dispositivos alimentados con baterías que no requieran de una recepción de información, como los nodos sensores, son de clase A.

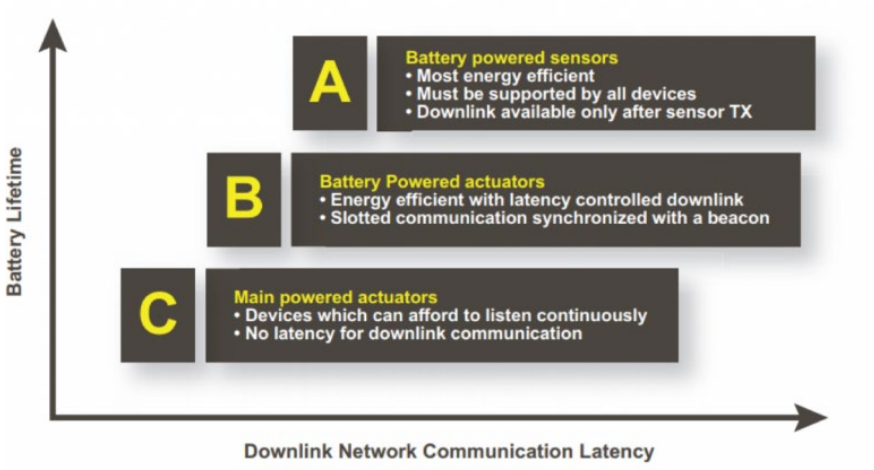

Figura 9. Consumo de energía de los dispositivos finales [24]

3.3. HARDWARE ELEGIDO

Hay una gran variedad de dispositivos que son compatibles con la tecnología LoRa, configurarlos a veces es más complejo y otras más sencillo, viniendo con la configuración básica y listos para ser usados. Para este proyecto se ha decidido utilizar los dispositivos de Heltec modelo WiFi LoRa 32 que permite las comunicaciones vía ondas de radio mediante la tecnología LoRa con los otros nodos y con el Gateway.

3.3.1. Módulo Heltec Wifi LoRa 32(V2)

Viene equipado con un microprocesador ESP32 y un chip de nodo LoRa SX1276/SX1278. Cuenta con un módem de largo alcance LoRa (distancia de comunicación abierta de 3km) y alta inmunidad a interferencias, minimizando el consumo de corriente. La frecuencia del módulo SX1278 es de 433MHz y 470MHz (China, Sudeste Asiático, Sudamérica y Europa del Este) y la del módulo SX1276 es de 868MHz y 915MHz (Europa y América del Norte). Por tanto, la frecuencia 868MHz es la que se va a usar para nuestras comunicaciones.

Además, incorpora el chip integrado CP2102 USB a puerto serie, para la descarga de programas sobre la placa y para la impresión de información de depuración. También cuenta con un display OLED de 128 * 64 de 0,96 pulgadas, que podemos utilizar para mostrar información en tiempo real.

La principal desventaja de este dispositivo es que su chip no implementa el protocolo LoRaWAN, que debe ser implementado posteriormente mediante librerías que faciliten estas capas de tecnología.

Para la implementación del sistema y realización de pruebas se han utilizado dos de estos nodos que actúan como sensores de información.

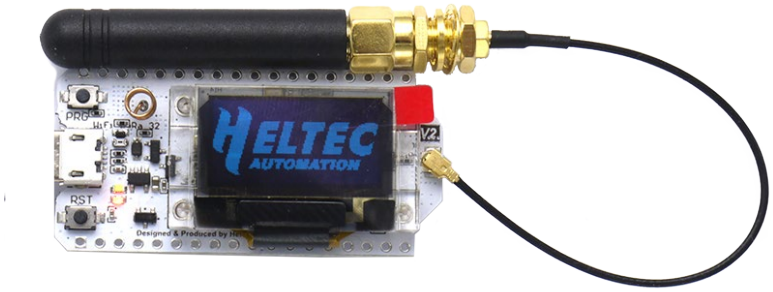

Figura 10. Heltec WiFi LoRa 32 (v2) [25]

3.3.2. Sensor BMP280

El sensor elegido para la obtención de datos ambientales es el sensor BMP280, que cuenta con librerías que facilitan su implementación y uso. Además, incluye ejemplos de utilización para ayudar al uso de sus funciones. Es un sensor barométrico de precisión con unas dimensiones compactas y un bajo consumo de energía, perfecto para proyectos de IoT.

Las métricas recogidas por el sensor son las siguientes:

- Rango presión: 300 ~ 1100 hPa
- Rango temperatura: -40 ~ 85ºC

Este dispositivo de medición va conectado mediante I2C o por SPI a nuestros nodos y les dota de la capacidad de recoger datos de presión barométrica, temperatura y estimar la altitud, que posteriormente serán enrutados por los nodos hasta llegar al servidor.

Figura 11. Sensor BMP280 [26]

3.3.3. Gateway RAK 2245 con Raspberry Pi 3B

El Gateway es uno de los dispositivos más importantes en la implementación. De él depende la zona de cobertura sobre la que van a comunicarse los nodos y el servidor y la capacidad de realizar comunicaciones simultáneas mediante sus canales.

La función principal de este elemento en la red es la de reenviar los mensajes enviados por los nodos con cobertura y el servidor, por lo que asegurarse de la compatibilidad entre extremos de la comunicación, es un punto importante a la hora de elegir el modelo.

El Gateway se comunica con el servidor de red mediante (3G/4G/5G), WiFi, Ethernet, fibra óptica o enlaces de radio de 2,4 GHz. Para la implementación en este proyecto se ha utilizado una red WiFi.

El modelo de Gateway que se usa para este proyecto lleva integrado un chip SX1301 para realizar las conexiones LoRaWAN y que pueda actuar de nodo Gateway LoRa, una antena que le permita una mejor recepción de paquetes y como sistema principal una Raspberry Pi.

Figura 12. Gateway utilizado en los experimentos

3.4. SERVIDOR DE RED

El servidor es el elemento que va a encargarse de gestionar el correcto funcionamiento de la red. Permitiendo la conectividad, gestionando y supervisando los dispositivos, puertas de enlace y aplicaciones, garantizando unos objetivos de seguridad, escalabilidad y fiabilidad. Por eso, es de suma importancia no solo la elección de un servidor adecuado, sino también su configuración, mantenimiento y seguridad.

Para la implementación del servidor en una red LoRaWAN, se debe asegurar que se cumplen las especificaciones características del estándar y así obtener un funcionamiento correcto. Debe ser capaz de eliminar mensajes duplicados, manejar tramas de enlace ascendente (Uplink) recibidas a través de los Gateway, tratar la capa MAC de LoRaWAN y programar transmisiones de datos de enlace descendente (Downlink).

A la hora de elegir servidor, es importante revisar los siguientes criterios para que cumpla con su correcto funcionamiento:

- Compatibilidad con los Gateway: El LNS (LoRaWAN Network Server) debe ser compatible con la gran mayoría de fabricantes, para evitar depender de un único desarrollador y no cerrarse ante nuevas soluciones.
- Solución robusta: Debe reducirse al máximo la pérdida de datos, que suele venir ocasionada por una pérdida de conexión con las puertas de enlace y la aplicación. Una solución ante esto, es que las propias puertas de enlace guarden la información, de tal forma que cuando la red local cae, la comunicación inalámbrica pueda suplir esa carencia.
- Consumo energético: El ADR (Adaptative Data Rate) es un valor utilizado para la optimización de transmisión de datos, que puede ayudar a aumentar la vida útil de las baterías de los dispositivos que componen la red.

- Gestión simplificada: Valorar la dificultad de utilización y despliegue de red LoRaWAN. Algunos ejemplos que la facilitan son la actualización automática de sensores y puertas de enlace, la monitorización de funcionamiento de los elementos, la capacidad de analizar la capa de red, permitir la decodificación de datos, el despliegue independiente de los otros elementos de la red, etc.

A continuación, se expone la opción elegida para actuar como servidor red de LoRaWAN, pero también se va a analizar dos opciones válidas para cumplir los estándares de red LoRa.

3.4.1 CHIRPSTACK

Se ha decidido la utilización de ChirpStack como servidor para la red LoRaWAN [27], antes conocido como Lora Server. Es uno de los más populares en este tipo de implementaciones open-source. Ofrece soporte para las redes LoRaWAN e incluye interfaces web intuitivas, además de dar soporte a aplicaciones externas mediante un API que permite su integración, ideal para proyectos de IoT. También permite configurar integraciones de datos con los principales proveedores de datos en la nube, bases de datos y servicios que se suelen utilizar para el manejo de datos en dispositivos, lo que añade la capacidad de ampliar las funcionalidades para proyectos más ambiciosos.

Algunas características de ChirpStack son:

- Soporte para las tres clases de dispositivos A, B y C.
- Tasa de datos adaptativa (ADR): Cuando esté habilitado, se asegura de que la velocidad de transmisión de datos y la potencia utilizada por los dispositivos sean lo más eficientes posibles, ahorrando batería, optimizando el uso del espectro de radio y minimizando la pérdida de paquetes.
- Registro de cuadros en vivo: Registra las tramas en vivo que se retrasmiten entre la puerta de enlace y el dispositivo, donde se puede apreciar el payload recibido, junto con los datos de la transmisión.
- Configuración de canales: Asegura la sincronización con los canales configurados en la red, independientemente de la utilización de subconjuntos de canales, o configuración de canales adicionales.
- Funciones Multiusuario: Permite la creación de perfiles de usuario y organizaciones que facilitan la visión de los distintos proyectos a los usuarios vinculados con ellos.
- API e integraciones: Mediante API gRPC y REST realiza un soporte para la integración de servicios externos.
- Compatibilidad con LoRaWAN 1.0 y 1.1, para todas las revisiones y bandas regionales de LoRaWAN.

3.4.2. THE THINGS NETWORK (TTN)

TTN es un servidor de red de código abierto compatible con todas las versiones de LoRaWAN [28]. Utilizado por una gran variedad de empresas y desarrolladores de todo el mundo, permite gestionar de forma segura aplicaciones, dispositivos finales y puertas de enlace.

Su arquitectura es robusta pero flexible, lo que le permite adaptarse a las propuestas más exigentes de LoRaWAN. Está formado por componentes débilmente acoplados que garantizan

seguridad, enrutamiento de datos y un uso óptimo de la batería de sus dispositivos finales, cumpliendo con el modelo de referencia de red LoRaWAN y asegurando el cumplimiento de sus estándares e interoperabilidad.

Una de sus mayores ventajas es la versión gratuita que ofrece, perfecta para el desarrollo de equipos y con un gran soporte de la comunidad. Sin embargo, para implementaciones más complejas será necesario su versión de pago. Cuenta con una interfaz intuitiva y la capacidad de integrar capas superiores, como conexión a bases de datos, APIs y otros servicios.

3.4.3. AMAZON WEB SERVICE (AWS) IoT CORE LORAWAN

Amazon ofrece una gran variedad de servicios que pueden facilitar el desarrollo y la implementación de sistemas informáticos, pero dentro de Amazon Web Service (AWS) encontramos el servicio AWS IoT Core [29], el cual nos permite conectar miles de dispositivos IoT de manera segura sin la necesidad de administrar la infraestructura. De hecho, Amazon proporciona un servidor de red completamente administrado para LoRaWAN.

AWS IoT Core para LoRaWAN es capaz de administrar las políticas de los dispositivos y servicios cuando se comunican con las puertas de enlace. Permite incorporar dispositivos LoRaWAN, puertas de enlace e incluso aplicaciones, sin la necesidad de gestionar servidores, únicamente realizando la configuración en cada uno de estos y conectándolos mediante las herramientas suministradas por Amazon.

4. PROPUESTA: REDES LORA MALLADAS

La propuesta presenta un sistema híbrido de recogida de datos basado en la tradicional arquitectura LoRaWAN y LoRa para crear redes LoRa malladas, que requieren de la obtención de diferentes tipos de datos, donde los nodos y Gateway están desplegados de forma estática. LoRaWAN es una solución de único salto, donde sólo los dispositivos dentro del rango de cobertura del Gateway pueden comunicarse con el servidor de red, esto limita el alcance de obtención de información únicamente al radio de actuación del Gateway. Además, pueden aparecer zonas de sombra causadas por la topología del terreno u obstáculos que impliquen la pérdida de datos. La implementación de una red mallada LoRa, pretende aportar una solución a los problemas de cobertura que pueden aparecer en despliegues reales utilizando redes LoRaWAN, respetando la filosofía de bajo consumo y largo alcance.

Mi propuesta va a permitir que aquellos nodos que no disponen de conexión directa con el Gateway utilizarán la red mallada de un único salto adicional para comunicarse con sus nodos más cercanos y descubrir cuál es el nodo con conexión directa al Gateway más fiable. Los nodos con conexión escucharán las comunicaciones LoRa de sus nodos cercanos, guardando todos los paquetes que reciben para después conectarse a la red LoRaWAN y enviar toda información, incluyendo la suya propia.

Esta solución, reduce la pérdida de información en las redes LoRaWAN, donde los nodos que no tengan conexión con el Gateway, por cualquier razón, pueden comunicar la información a los nodos cercanos que sí posean conexión. También, genera la posibilidad de utilización de nodos sensores fuera de las zonas de cobertura, para una obtención de datos más amplia y que abarque un mayor terreno, sin la necesidad de desplegar Gateways adicionales.

4.1. ARQUITECTURA

En las comunicaciones LoRa de la red mallada, al realizar un envío de paquetes, se hace una difusión a todos los nodos en el rango. Sin embargo, se ha diseñado un protocolo que define la forma de actuación para los nodos, cuando estos reciben un paquete, permitiendo únicamente las comunicaciones de un salto entre nodos con conexión y nodos sin conexión e ignorando el resto de las comunicaciones que se reciban en los dispositivos (ver Figura 13). Una red mallada de más de un salto generaría una gran cantidad de tráfico en el espectro de radiofrecuencia, por encima de las recomendaciones de LoRa y en contra de su filosofía.

El funcionamiento está claramente dividido en dos fases:

- Fase 1: Está fase se ha denominado “Discover”. Aquí el comportamiento de los nodos se va a dividir en función de si pueden establecer conexión con el Gateway, o no. Una vez los nodos saben si tiene conexión o no, pasan a usar las comunicaciones LoRa P2P. En concreto, los nodos que no tengan conexión con el Gateway van a comunicarse con los nodos que estén a un único salto para descubrir si tienen conexión, mediante el envío de paquetes “discover”. Se reciben las respuestas de los nodos que dispongan de conexión y se elige el nodo final más fiable basándose en su señal RSSI, estableciéndolo como el nodo destino al que enviar su información.

- Fase 2: Una vez los nodos sin conexión tienen asignado un nodo destino, le envían la información captada por sus sensores en periodos de tiempo establecidos. Por cada paquete enviado al destino, comprueban si el nodo destino sigue disponible, es decir, si sigue teniendo conexión con el Servidor y, de no estar disponible, se ejecuta de nuevo la fase Discover. Los nodos con conexión directa al Gateway escuchan durante un periodo de tiempo los paquetes de los nodos cercanos mediante las comunicaciones LoRa. Responden a los paquetes discover que reciben y guardan los paquetes recibidos con la información de otros nodos sensores. Cuando ha pasado este periodo de escucha, los nodos inician las comunicaciones LoRaWAN y envían la información recogida, junto con la que ellos han captado, al servidor de ChirpStack.

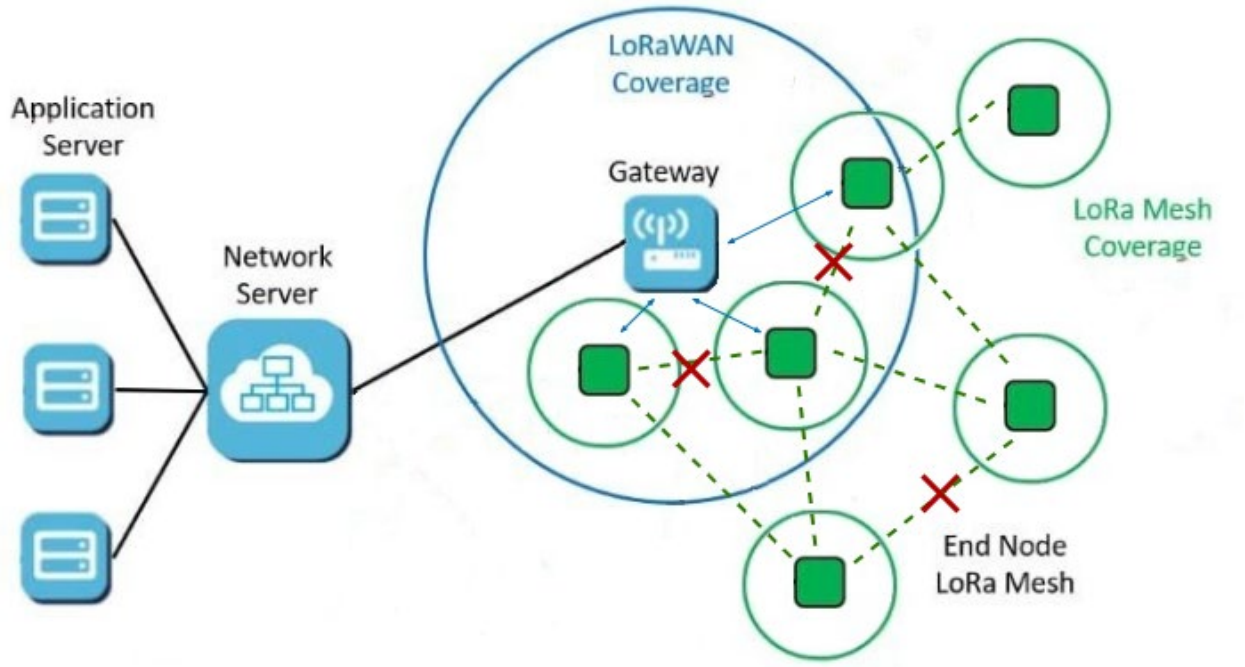

Figura 13. Arquitectura comunicaciones LoRaWAN y LoRa Mesh

4.2. ENVÍO DE LA INFORMACIÓN

Antes de realizar operaciones de envío entre dispositivos LoRa, se debe definir el contenido que va a ir empaquetado en estos mensajes. Para garantizar la seguridad en la red mallada y realizar una buena gestión de las comunicaciones, es necesario la identificación del origen y destino de paquetes, por lo que esta información va siempre presente en cada paquete enviado en la red. En esta sección se detalla el proceso de identificación de los dispositivos, los métodos de empaquetado y desempaquetado y la estructura de los diferentes tipos de paquetes que se envían.

4.2.1. IDENTIFICACIÓN DE DISPOSITIVOS

El id de los dispositivos debe ser único y juega un papel esencial a la hora de gestionar las comunicaciones con otros nodos, ayudando a filtrar las comunicaciones e identificando si los paquetes recibidos van dirigidos a él o deben ignorarse, es decir, tienen una gran importancia a la hora de diseñar la lógica de la red LoRa mallada.

Se ha decidido usar como identificación única la clave DevEUI generada por el servidor ChirpStack en el proceso de activación y registro de dispositivos. La clave DevEUI generada en formato hexadecimal tiene un tamaño de 8 bytes y se guarda en un array de bytes llamado localAddress.

4.2.2. ESTRUCTURA DE PAQUETES

Se han implementado funciones de empaquetado y desempaquetado de datos. La función de empaquetado transforma los datos que componen el paquete en un array de bytes en formato hexadecimal. La función de desempaquetado realiza lo contrario, obteniendo los valores originales de la ristra de bytes recibida y guardándolos en variables, permitiendo gestionar la

información. Es importante que estos paquetes no superen la cantidad máxima de 255 bytes que pueden enviar los módulos LoRa.

Los identificadores de dispositivos ya están en formato hexadecimal, sin embargo, los datos captados por los sensores deben ser convertidos a este formato. Para este TFG, se ha captado la temperatura mediante sensores, a modo de ejemplo, para añadirla como información a los paquetes. Esta temperatura en formato float es convertida un número hexadecimal de 4 bytes mediante la función que se muestra en la Figura 14.

```
void convertFloatToHex(unsigned char* output, float floatVariable){
  unsigned char Array[4] = {
      ((unsigned char*)&floatVariable)[0],
      ((unsigned char*)&floatVariable)[1],
      ((unsigned char*)&floatVariable)[2],
      ((unsigned char*)&floatVariable)[3]
   };
  memcpy(output,Array, sizeof(Array));

}
```

Figura 14. Función conversora Float-Hexa

Durante las comunicaciones P2P LoRa, el tamaño mínimo de los paquetes con los que se comunican los dispositivos es de 16 bytes. Estos conforman la estructura básica de un paquete, donde los primeros 8 bytes son el identificador del nodo destino del mensaje y los 8 últimos el identificador del nodo que lo envía. Estos paquetes son usados principalmente para enviar peticiones de descubrimiento de nodos y para realizar respuestas. Para enviar información recogida por los dispositivos, sobre estos paquetes básicos se añadirán los datos recogidos por los sensores que sean necesarios enviar, siempre que no sobrepasen el máximo de 255 bytes permitidos. Cuando los nodos reciben los paquetes, son conocedores de la estructura y el orden de la información.

Como se ha comentado, en las comunicaciones LoRa se envían distintos paquetes. Estos se clasifican en 3 categorías diferentes según su función:

- **Paquetes discover**: Estos paquetes son enviados en el proceso discover, a los nodos cercanos, con la finalidad de obtener una respuesta de aquellos nodos que dispongan de una conexión directa con el Gateway. De esta forma pueden ser establecidos como destino al que reenviar la información.
  El paquete tiene un tamaño de 16 bytes, los 8 primeros son para identificar al paquete y los 8 últimos contienen el identificador del nodo origen. Como se puede observar en la Figura 15, estos paquetes son identificados por sus primeros 8 bytes, donde el destino viene definido por 8 bytes con valor hexadecimal 0xFF. De esta forma, los nodos que reciben el paquete pueden identificar que se trata de un paquete discover.

Figura 15. Destino de paquetes discover

- **Paquete respuesta**: Utilizado para confirmar la recepción de un paquete, a modo de ACK. Los nodos con conexión directa al Gateway responden al nodo remitente con estos paquetes cuando reciben paquetes discover, para confirmar la entrega de datos en el servidor a los nodos que están pendientes del resultado.
  El paquete tiene un tamaño de 16 bytes, los 8 primeros contiene el identificador del nodo al que va dirigida la respuesta y los 8 últimos el identificador del nodo origen (Ver Figura 16).

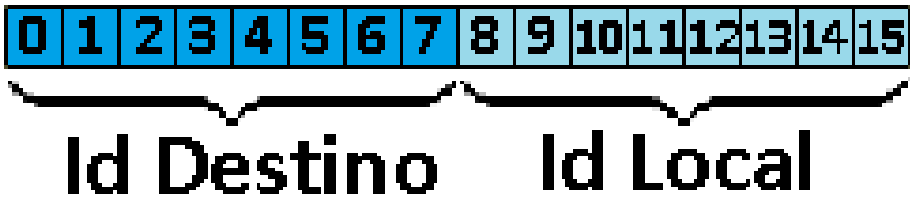

Figura 16. Estructura paquete respuesta

- **Paquetes de datos:** Los nodos sin conexión al Gateway, una vez han descubierto un camino disponible, captarán la información de sus sensores y la empaquetarán junto con los demás datos para enviársela al nodo que han asignado como destino.
  Este paquete tiene un tamaño máximo variable, dependiente de la cantidad de datos que se añadan. Sin embargo, todos los paquetes van determinados por los mismos 16 primeros bytes. Los 8 primeros bytes contienen el identificador del nodo destino, los 8 siguientes contienen el identificador del nodo origen, y los últimos contienen los datos captados (Ver Figura 17).

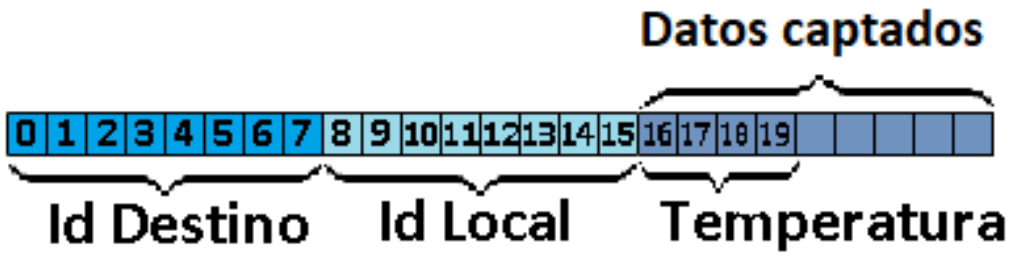

Figura 17. Estructura de paquetes con datos

Finalmente, para la comunicación con la red LoRaWAN, se utiliza un cuarto tipo de paquete.

- **Paquete servidor:** Este paquete contiene la información que se envía al servidor de ChirpStack desde los nodos conectados. Contiene en los 8 primeros bytes el identificador del nodo que captó la información que lleva el paquete. Tras los 8 primeros bytes se almacenan los datos recogidos por ese nodo, que como ya se ha explicado pueden variar (Ver Figura 18). Se guarda en una lista todos los paquetes servidor, derivados de los paquetes con datos recibidos de otros nodos, e incluyendo los datos captados por el propio sensor del nodo.

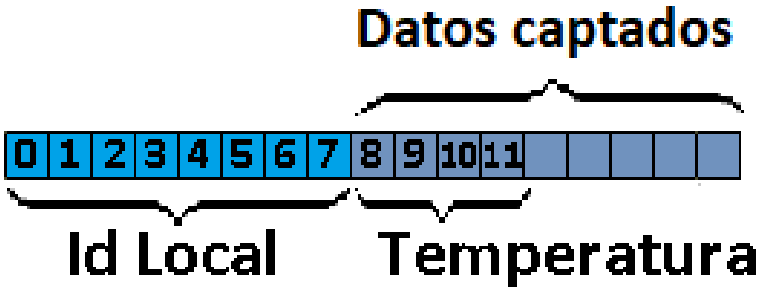

Figura 18. Estructura de paquetes servidor

4.2.3. EMPAQUETADO

Como se ha comentado, en las comunicaciones LoRa se envían paquetes compuestos por tres datos: destino (8 bytes), id local (8 bytes) y datos del sensor (bytes indefinidos), todas guardadas en variables tipo byte array. Para los identificadores de destino y origen de los paquetes la implementación en código es fácil, debido a su tamaño fijo, ya que se define de manera inicial el tamaño que se almacena en estas variables byte array. Por lo tanto, para empaquetar estos dos valores de tamaño fijo, simplemente hay que copiarlos en una nueva variable con el tamaño total de ambos. Sin embargo, para dar la capacidad de añadir diferentes datos sensores a los paquetes, y que la cantidad de bytes sea variable, se ha diseñado otro sistema.

Para guardar los datos de sensores, se ha creado una lista de estructuras (Ver Figura 19) que contiene un puntero a la variable donde se va a añadir cada dato que se quiere empaquetar en el mensaje. Cada vez que se obtiene información para añadir al paquete, se crea un nuevo elemento de tipo estructura en la lista con la información almacenada en la variable “data”. Finalmente, cuando se acaba de añadir los distintos tipos de datos a la lista, se recorre entera, añadiendo la información almacenada en la dirección de “data” de cada elemento de la lista a un nuevo byte array (Ver Figura 20), creado con el tamaño de bytes totales de datos captados que contiene la lista. Una vez se tiene toda la información guardada en la variable “dataSensor”, y conociendo así el tamaño de bytes de datos que se quiere enviar, se empaqueta esta información después de los 16 primeros bytes de identificación.

Este mismo sistema se utiliza también para almacenar los paquetes servidor. Cuando un nodo con conexión recibe un paquete con datos**,** empaqueta el identificador del nodo que lo ha enviado junto con la información que ha captado, añadiéndolos a la lista en la dirección de la variable “data”. En esta nueva instancia de la lista se guardan todos los paquetes que más tarde se envían al servidor.

Este sistema de almacenamiento de datos dinámico, permite el almacenamiento heterogéneo de valores sin importar el tipo de dato captado por el sensor. Sin embargo, requeriría de la implementación de funciones especificas para cada tipo de dato que transformen el contenido a hexadecimal. Después, estos datos se añaden a la posición correspondiente en la lista, junto a los demás datos captados.

```
typedef struct
  {
      int size;
      byte* data;
  }  packetData;
packetData* sensorDataList;
packetData* packetList;
```

Figura 19. Almacenamiento de información variable

```
byte* dataSensor;
```

Figura 20. Puntero a la variable con los datos captados en una ristra de bytes

Se declaran las listas y datos mediante punteros, porque no se sabe desde un principio ni cuántos elementos de información se van a añadir, ni de qué tamaño van a ser, por lo que en el momento que se recogen los datos se define su tamaño.

La lista con los datos dispone de tres funciones:

- Creación de lista: Inicialmente se crea una lista con un tamaño de 10 elementos como máximo, pero mediante el uso de las otras funciones, este tamaño se podría aumentar.
- Añadir elemento: Añade un elemento del tipo de la estructura definida, que contiene un tipo de datos captado y el tamaño que ocupa en bytes.
- Aumentar tamaño: Esta función se llama al llegar al límite definido en un principio. Lo que hace, es copiar el contenido de la lista actual y crear una nueva con mayor tamaño a la que añade el contenido de la anterior (Ver Figura 21).

```
void resizeList(packetData* list, int sizeList, size_t newCapacity)
{
  packetData* newList = new packetData[newCapacity];
  memmove(newList, list, sizeList);
  delete[] list;
  list = newList;
}
```

Figura 21. Función de aumento de tamaño de la lista

4.2.4. DESEMPAQUETADO

Al recibirse un paquete, se procede a leer la información que contiene para actuar de una manera determinada. Como los nodos son conocedores de la estructura básica de los paquetes, se tiene constancia de que siempre los primeros 16 bytes pertenecen a la identificación de destino y a la identificación del nodo que ha enviado el paquete, de 8 bytes cada uno y en ese orden. Por lo tanto, si el paquete ocupa más de estos 16 bytes, significa que se ha recibido un paquete con datos y que los bytes restantes del paquete recibido corresponden a la información recogida por el nodo que ha enviado el paquete. Una vez desempaquetada la información conociendo este orden y tamaño del paquete, se decide el comportamiento del programa en torno a ella.

```
while (LoRa.available())
{

  if( i < sizeof(destino)){
    destino[i] = LoRa.read();
    Serial.print(destino[i],HEX);
    Serial.print(":");

  }
  if (i >= sizeof(destino) && i < (sizeof(sender) + sizeof(destino))){
    sender[j] = LoRa.read();
    Serial.print(sender[j],HEX);
    Serial.print(":");
    j++;
  }

  if(i >= (sizeof(sender) + sizeof(destino))){
    sensorInfo[j] = LoRa.read();
    Serial.print(sensorInfo[j],HEX);
    Serial.print(":");
    j++;
  }
  i++;
}
Serial.println();
Serial.println("Fin paquete ");
```

Figura 22. Función desempaquetado LoRa

4.3. COMUNICACIONES LORA

A partir de los conceptos básicos de comunicación mediante LoRa, se ha definido las funciones y comportamiento básico de la red mallada. Como ya se ha indicado anteriormente, las transmisiones LoRa se realizan en broadcast y pueden ser recibidas por los dispositivos que están a su alcance. Para el enrutamiento de paquetes hacia dispositivos concretos nos ayudamos de los identificadores únicos de cada nodo LoRa, empaquetando esta información en cada mensaje enviado. Gracias a la identificación del destino y el origen del paquete, podemos filtrar las comunicaciones, indicando cuáles se deben ignorar y cómo actuar en base a esta información. Para el envío de paquetes se ha desarrollado la función sendMessage() y para la recepción, la función onReceive(). Ambas funciones utilizan la librería LoRa.h para la implementación de funciones de comunicación P2P LoRa.

4.3.1. ENVÍO DE MENSAJES

SendMessage() es una función sencilla para el envío de paquetes, mediante LoRa, a dispositivos dentro del alcance. Se le pasa por parámetro la información a enviar, ya empaquetada, junto con el tamaño que ocupa. La implementación de esta función en Arduino se puede observar en la Figura 23. Este paquete será enviado a todos los nodos LoRa

```
void sendMessage(uint8_t* payload, int size)
{
  LoRa.beginPacket();
  LoRa.write(payload, size);
  LoRa.endPacket();
  Serial.print("Enviando mensaje: ");
  for(int i = 0; i< size; i++){
    Serial.print(payload[i],HEX);
    Serial.print(":");
  }
}
```

Figura 23. Función para envío de mensajes en LoRa mesh

4.3.2. RECEPCIÓN DE MENSAJES

La función onReceive() comprueba si se recibe un paquete y define el comportamiento del dispositivo ante los mensajes recibidos. Cuando se recibe un paquete, lo desempaqueta y lee todo el contenido para identificar el destino y el origen de dicho mensaje. Después, se definen dos comportamientos distintos, dependiendo de si este nodo tiene conexión directa con el Gateway o no. Para ayudar a la explicación en la Figura 24 se representa el comportamiento de la función onReceive() mediante un diagrama de actividad.

En los nodos que tiene conexión con el Gateway, solo se espera recibir paquetes discover y paquetes con datos de los nodos sin conexión asociados, por lo que se van a ignorar el resto de las comunicaciones recibidas:

- Ha recibido un **paquete discover**: El destino del paquete viene identificado por los bytes de un paquete discover. Significa que el nodo que lo ha mandado está buscando nodos con conexión directa al Gateway. Se empaqueta un nuevo mensaje con la respuesta y envía el paquete respuesta con el identificador de ese nodo como destino.
- Ha recibido un **paquete con datos**: El destino del paquete coincide con el identificador del nodo que lo recibe, por lo que el mensaje ha llegado a su destino y se trata de un paquete con datos, ya que solo los nodos sin conexión envían paquetes con datos a sus nodos destino asociados. Se obtiene el identificador del nodo origen (8 bytes) y los datos captados, en la que empaquetan ambos, formando un paquete servidor. Este paquete se añade la lista (Ver Figura 19*)* donde se almacenan los paquetes que se van a enviar al servidor.

En cada uno de los nodos que no disponen de conexión al Gateway, se ignoran todos los paquetes que no van destinados a él, es decir, los 8 primeros bytes del paquete que identifican el destino no coinciden con el identificador del nodo que lo recibe. Esto se hace para ignorar las comunicaciones con los demás nodos, pues solo se esperan las respuestas de los nodos que

tienen conexión con el Gateway, a los cuales se ha contactado previamente. Los paquetes respuesta recibidos pueden deberse a dos motivos en función del origen del paquete:

- Respuesta a un **paquete discover:** Se recibe un paquete respuesta, enviado por un nodo con conexión al Gateway en contestación al paquete discover enviado por este nodo con anterioridad. Se guarda la identificación del nodo origen que ha enviado el paquete junto con el nivel de RSSI de esta transmisión en una lista. La implementación del guardado de estos dos datos asociados en Arduino se muestra en la Figura 25.
- Confirmación de **paquete con datos**: Si el paquete recibido proviene del nodo destino que tiene asociado, significa que es un paquete respuesta**,** que confirma que el paquete con datos enviado anteriormente ha sido entregado al servidor ChirpStack y, que este destino sigue estando disponible para seguir enviándole paquetes con datos.

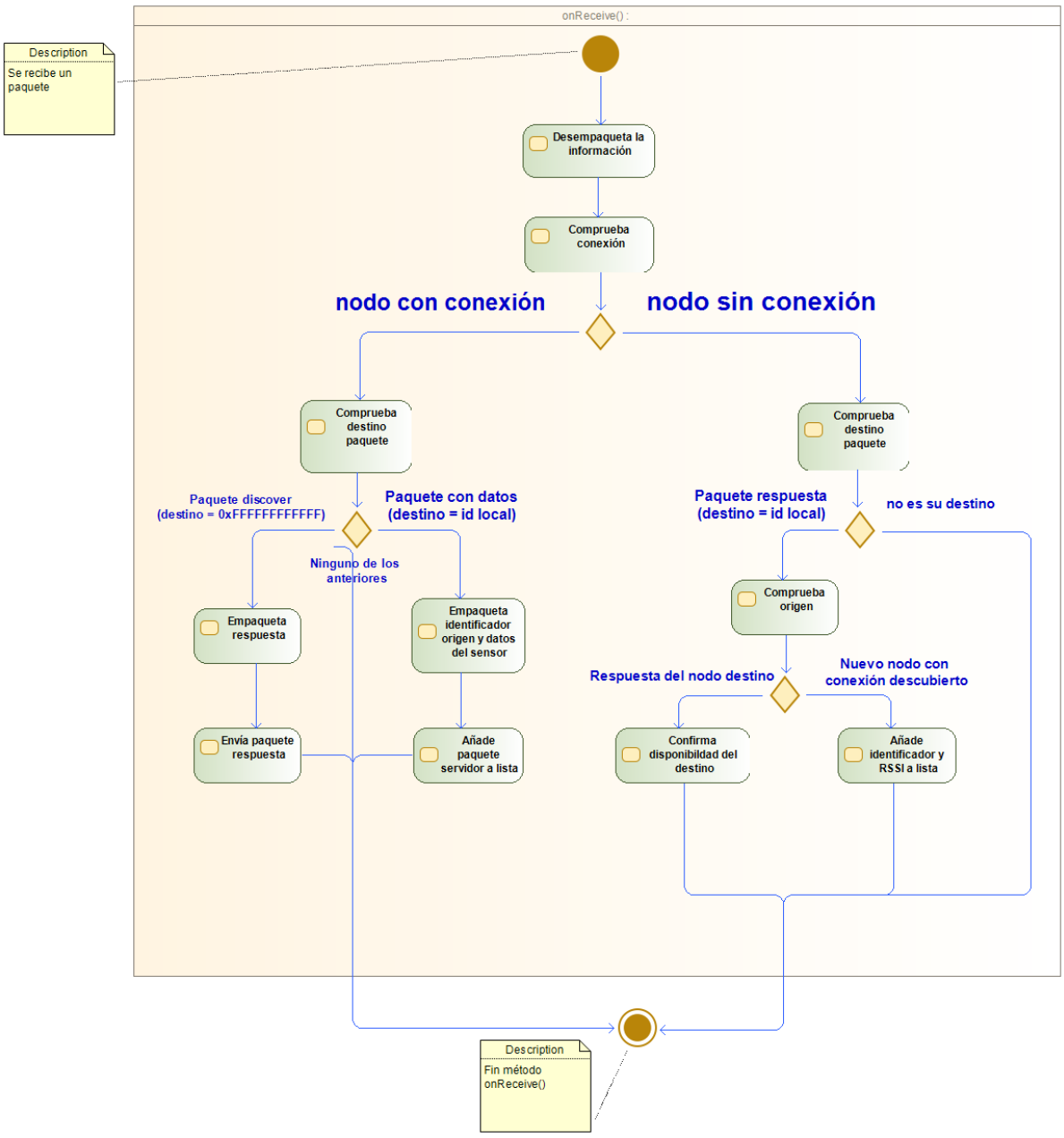

Figura 24. Diagrama de actividad función onReceive()

```
typedef struct
  {
      byte idNodo[6];
      int rssi;
  }  nodoDescubierto;

nodoDescubierto listaNodos[NUM_NODOS];
```

Figura 25. Implementación de lista de nodos destino descubiertos

4.3.3. RECHAZO DE MENSAJES DESCONOCIDOS

Cuando un nodo envía un mensaje, lo difunde a todos los nodos dentro de su rango y bajo su misma frecuencia, siempre que usen comunicación LoRa. Esto quiere decir que nuestros mensajes pueden llegar a desconocidos, comprometiendo la seguridad de la red mallada.

Para evitar que esto suceda se usa la función ***LoRa.setSyncWord(0xF3)***, con la que podemos establecer una palabra de sincronización con un rango de 0 hasta 0xFF. Se asigna una misma palabra de sincronización para todos los nodos de nuestra red y asegura que solo se envíen y reciban comunicaciones con otros dispositivos LoRa con la misma palabra de sincronización.

4.4. COMUNICACIONES LORAWAN

Para que los dispositivos finales formen parte de la red LoRaWAN y puedan comunicarse con el servidor, se ha realizado previamente la activación de elementos que conforman la red LoRaWAN, como se ha visto anteriormente en el apartado de configuración del servidor ChirpStack y se implementan en Arduino una serie de funciones que facilitan la comunicación en la red LoRaWAN.

4.4.1. Configuración de claves

Una vez generadas las claves DevEUI y AppKey para el dispositivo final en ChirpStack, se definen dentro del código en el dispositivo, de manera que se pueda establecer conexión con el Gateway y el servidor, comenzando las comunicaciones dentro de la red LoRaWAN. Además, se utiliza también esta clave DevEUI para la identificación de nodos.

4.4.2. Establecer conexión con el servidor

Mediante la función join() de la librería LoRaWAN, los dispositivos se comunican con el Gateway para establecer una sesión con el servidor de red ChirpStack. Esta función nos devolverá un booleano, resultado de la conexión con el servidor. Mediante este resultado podemos guiar el comportamiento de los nodos al saber si están dentro o fuera del rango de actuación del Gateway.

4.4.3. Envío de datos

Si la conexión se realiza con éxito, los dispositivos finales pueden comenzar con el envío de paquetes al servidor por medio de Uplinks. En Arduino se implementa la función sendToGateway(), a la cual se le pasan los paquetes servidor**,** junto con su tamaño en bytes. El nodo final realiza un envío de Uplink por cada paquete servidor almacenado en la lista de paquetes que ha recibido de otros nodos.

```
void sendToGateway(uint8_t* payload, int size){
  Serial.println("Sending data to Server");

  lora.sendUplink((char* )payload, size, 0, 1);

  recvStatus = lora.readData(outStr);
  if(recvStatus) {
    Serial.println(outStr);
  }

  lora.update();
}
```

Figura 26. Función LoRaWAN para envío de datos al servidor

4.4.4. Visualización de payload

En el servidor ChirpStack aparecen las conexiones entrantes de los dispositivos finales, así como los Uplinks recibidos. Por defecto, no se puede visualizar en la interfaz de ChirpStack el payload de los paquetes, ya que vienen cifrados. Se implementa pues, un códec en leguaje JavaScript que realiza la decodificación del payload recibido, el cual, contiene la información del paquete servidor que se ha subido. Decodifica los 8 primeros bytes correspondientes al identificador del nodo que ha captado la información, y los siguientes bytes correspondientes a la información.

En esta implementación, como se ha usado la temperatura captada por los sensores como demostración, transforma la ristra de 4 bytes de la temperatura a tipo float para poder mostrar la información del payload en la interfaz de ChirpStack. En la Figura 27, se muestra un paquete servidor codificado y, decodificado, con el identificador del nodo y la temperatura en ºC.

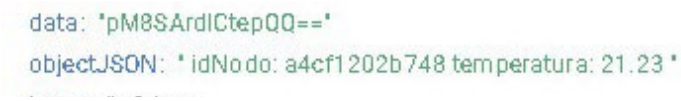

Figura 27. Ejemplo de datos recibidos en ChirpStack (codificados y decodificados)

4.5. AHORRO DE ENERGÍA

No se necesita que los dispositivos estén constantemente enviando y recibiendo información. Así que, para optimizar la vida útil de las baterías y que duren más tiempo, se va a optimizar el uso de la energía en estos dispositivos. En los intervalos de tiempo que no se requiera la comunicación entre dispositivos, en vez de permanecer activos, estos se ponen en suspensión profunda donde no se llevan a cabo actividades de CPU o WI-Fi. La memoria RTC (Real Time Controller) permanecerá encendida y el coprocesador ULP (Ultra Low Power) aún puede encenderse. La finalidad de esto es minimizar el uso de recursos y aumentar la duración de sus baterías.

Para ello, se ha implementado el uso de la función esp_deep_sleep() y así ahorrar energía. Dicha función configura los dispositivos para que se suspendan durante el tiempo definido. Los nodos comenzarán a hibernar cuando hayan cumplido con su ciclo de ejecución y después reiniciará el dispositivo para realizar un nuevo ciclo.

4.6. USO DE LA MEMORIA PERSISTENTE

Para guardar el estado del programa, ante posibles reinicios o cortes de energía, se utiliza la memoria persistente para almacenar las diferentes variables que permitan al programa recuperar el estado que tenía antes de apagarse así evitar tener que realizar los procedimientos de nuevo.

Se ha utilizado las funciones de la librería EEPROM para leer y escribir la memoria flash del ESP32, que nos permite usar 512 direcciones diferentes, con un valor entre 0 y 255. Se configura la cantidad de bytes que se usan al inicializar la EEPROM.

En la implementación se reservan 18 bytes donde se almacenan los siguientes valores, por este orden:

- [0] hayValor: Se almacena una variable de tipo booleano para identificar si previamente se había añadido algún valor a la EEPROM.
    - Si el valor de esta variable es igual a 1, significa que ya se han metido valores en la memoria EEPROM en una ejecución anterior, y por lo tanto, cuando se inicie el nodo se restaurarán los valores.
    - Si el valor de esta variable es distinto de uno, cumplirá con la ejecución normal, realizando el proceso estándar y obteniendo los valores. Al acabar una ejecución completa del programa, escribirá los valores en la EEPROM e indicará que se ha escrito sobre ella, poniendo el valor 1 en la dirección 0 de la memoria.
- [1] conexiónGateway: Almacenar una variable booleana que guarda si el dispositivo final tiene conexión directa con el Gateway.
- [2-9] localAddress: Almacena los 8 bytes de identificación del nodo.
- [10-17] destination: Almacena los 8 bytes de identificación del nodo que tiene asignado como destino. Cuando se encuentra un nuevo destino se sobrescribe esta región de la memoria con el identificador del nuevo nodo encontrado.

4.7. PROTOCOLO DISCOVER

Protocolo que implementan los nodos sin conexión para descubrir el camino disponible más fiable hacia el Gateway mediante las comunicaciones LoRa, y al que retransmiten el tráfico de sus paquetes con datos.

Función discover()

Se ha implementado una función discover() (Ver Figura 28), que envía un paquete discover a los nodos que estén a su alcance. Después utiliza la función onReceive() durante un periodo de tiempo establecido para escuchar las comunicaciones y, si no se ha recibido ningún paquete respuesta destinado a este nodo, volverá a ejecutar el proceso de envío y escucha hasta recibir como mínimo una respuesta. Cuando recibe un paquete respuesta para este nodo, se mide la señal RSSI de la transmisión y se añade a una lista, junto con el identificador del nodo que ha respondido. Una vez pase el tiempo de escucha y reciba como mínimo una respuesta, se recorre la lista entera de valores RSSI y se elige el nodo con mayor RSSI, definiéndolo como el destino de las comunicaciones. Estos 8 bytes que contiene la identificación del nuevo nodo destino, son escritos en la memoria EEPROM en las direcciones [10-17].

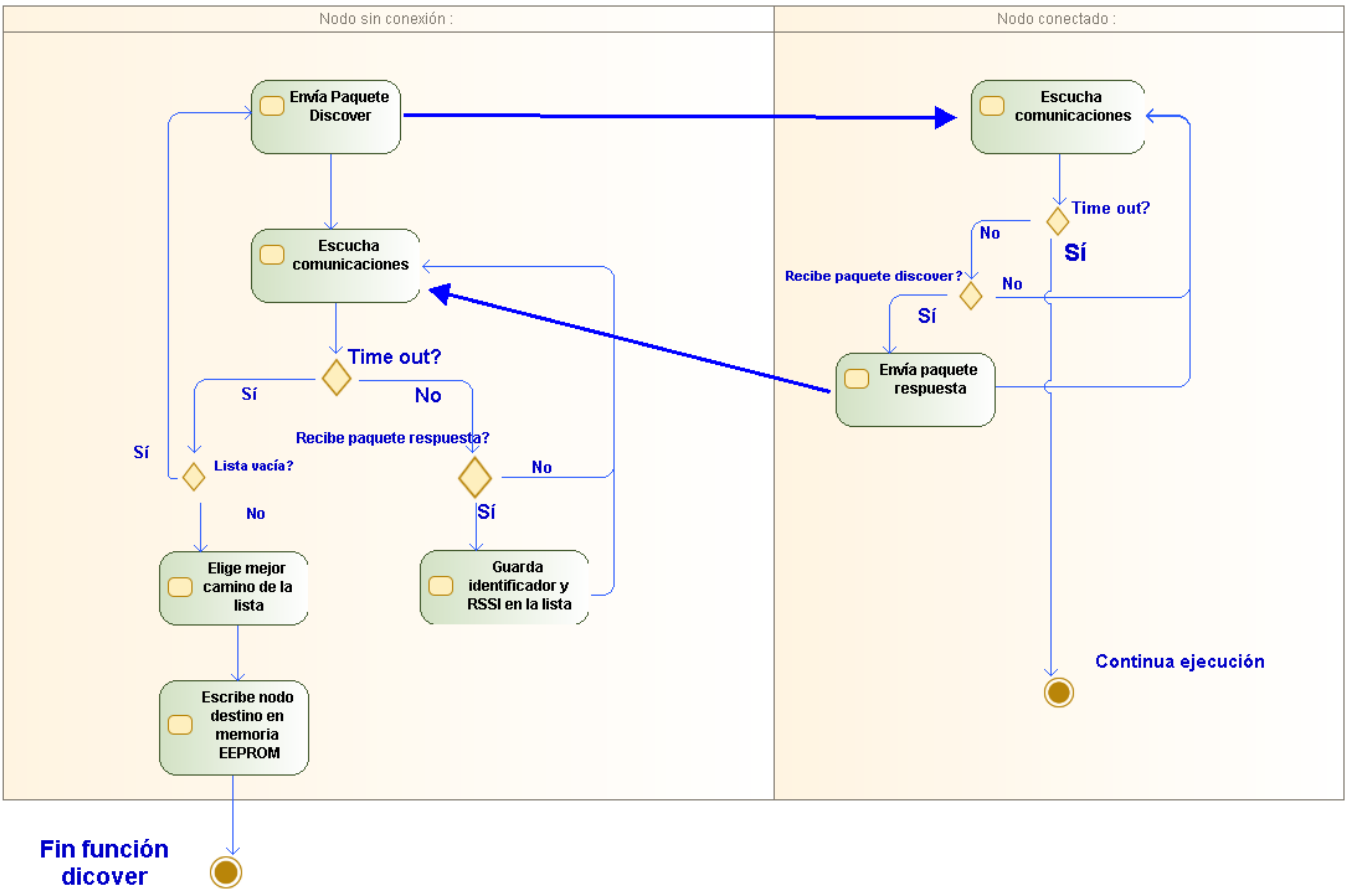

Figura 28. Diagrama de actividad función discover()

Confirmación disponibilidad del destino

Otra de las funciones de este protocolo discover, es la capacidad de detectar cuando un destino ha dejado de estar disponible. Mediante el uso de paquetes respuesta se confirma la recepción de la información en el servidor. Una vez se ha descubierto un nodo destino, se le redirigen los paquetes con datos captados. Después de cada envío de paquete con datos, se queda escuchando durante un periodo de tiempo una respuesta por parte del nodo destino, que le confirme que el paquete ha sido recibido en el servidor. Si el tiempo de escucha llega a su fin y no se recibe ningún paquete respuesta para este nodo, este destino se dará por inalcanzable, borrando su identificador como destino del nodo, y se volverá a ejecutar la función discover() para encontrar un nuevo nodo destino disponible.

El nodo destino, durante el periodo de escucha, puede recibir aquellos paquetes con datos que le van dirigidos. Cuando recibe un paquete con datos, guarda el identificador del nodo origen y la temperatura en un nuevo paquete servidor y lo añade a la lista donde se almacenan todos los paquetes que se van a enviar al servidor. Cuando finaliza el periodo de escucha, el propio nodo obtiene, si tiene, los datos de su propio sensor y crea un nuevo paquete servidor para la lista. Finalmente, comprueba si hay algún paquete en la lista para enviarlo al servidor. Si hay paquetes, se cierra la conexión LoRa y se inicia LoRaWAN, conectándose a la red mediante sus credenciales. Una vez conectado, recorre la lista y realiza un Uplink al servidor con cada paquete servidor. Después de enviar todos los paquetes, cierra la comunicación LoRaWAN e inicia de nuevo LoRa. Recorre la lista y envía un paquete respuesta a todos los nodos que están en ella, para confirmar que se han enviado los paquetes con datos al servidor.

Este sistema tipo ACK de confirmación de paquetes recibidos, también es utilizado para la sincronización de las comunicaciones P2P entre dispositivos, debido a los métodos de hibernación implementados. Una vez los nodos con conexión han enviado todos los paquetes respuesta, entran en hibernación. Los nodos sin conexión a la espera de un paquete respuesta, hibernarán una vez lo reciban. De esta manera, se realizan los ciclos de hibernación y transmisión de paquetes en los mismos intervalos de tiempo.

Se pensó una solución más sencilla y eficaz para la sincronización según la hora actual, definiendo los ciclos de hibernación de los dispositivos en tiempos concretos según la hora local. El problema de esto es la necesidad de que los dispositivos lleven integrado GPS. A la hora de la implementación los nodos Heltec WiFi LoRa 32 no disponían de GPS y se diseñó la solución explicada en esta sección para la sincronización de los dispositivos.

En la Figura 29 puede observarse el comportamiento anteriormente descrito por ambos nodos.

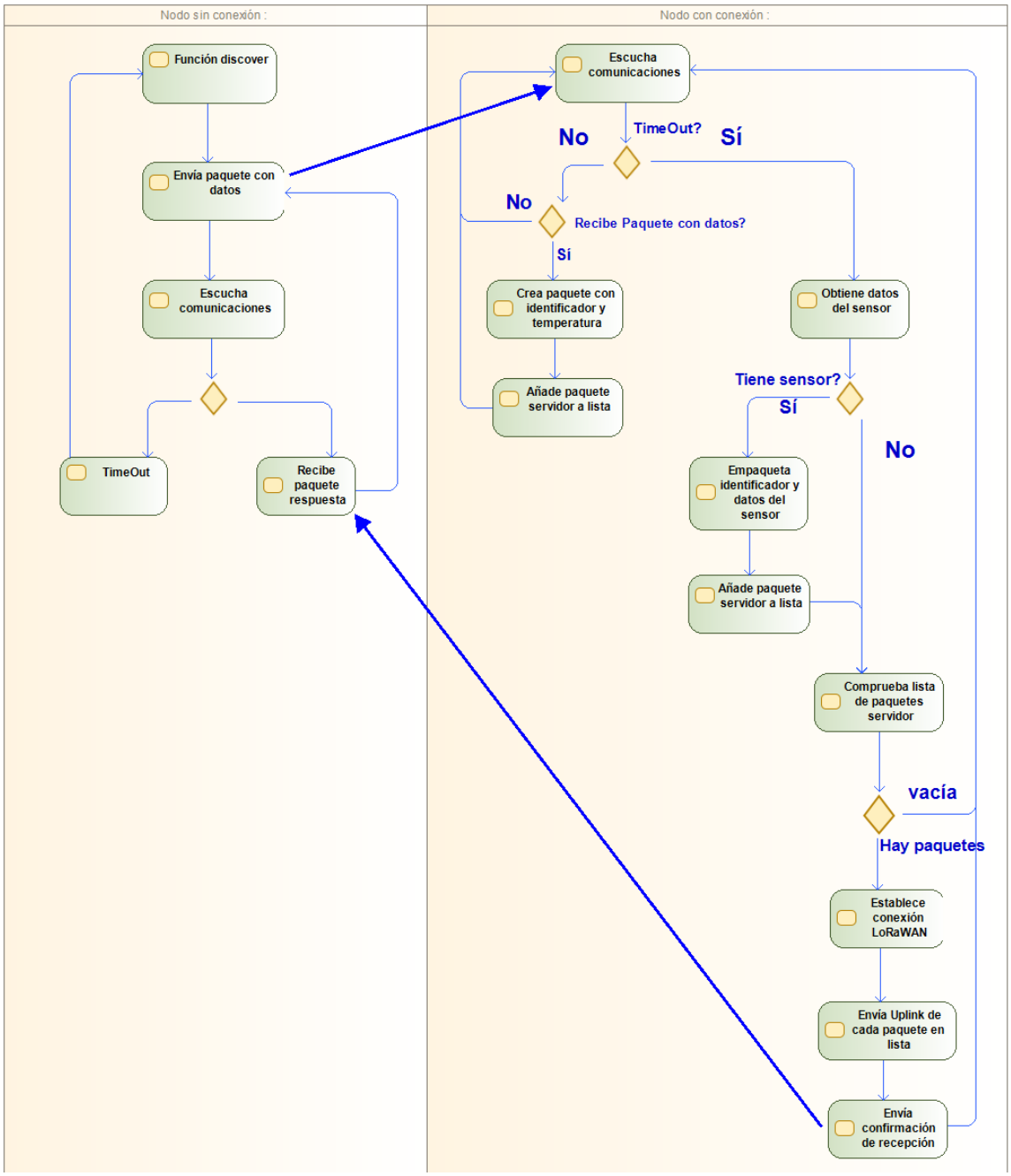

Figura 29. Diagrama de actividad disponibilidad de destino

4.8. DIAGRAMA DE ACTIVIDAD DE LA LÓGICA DEL PROGRAMA

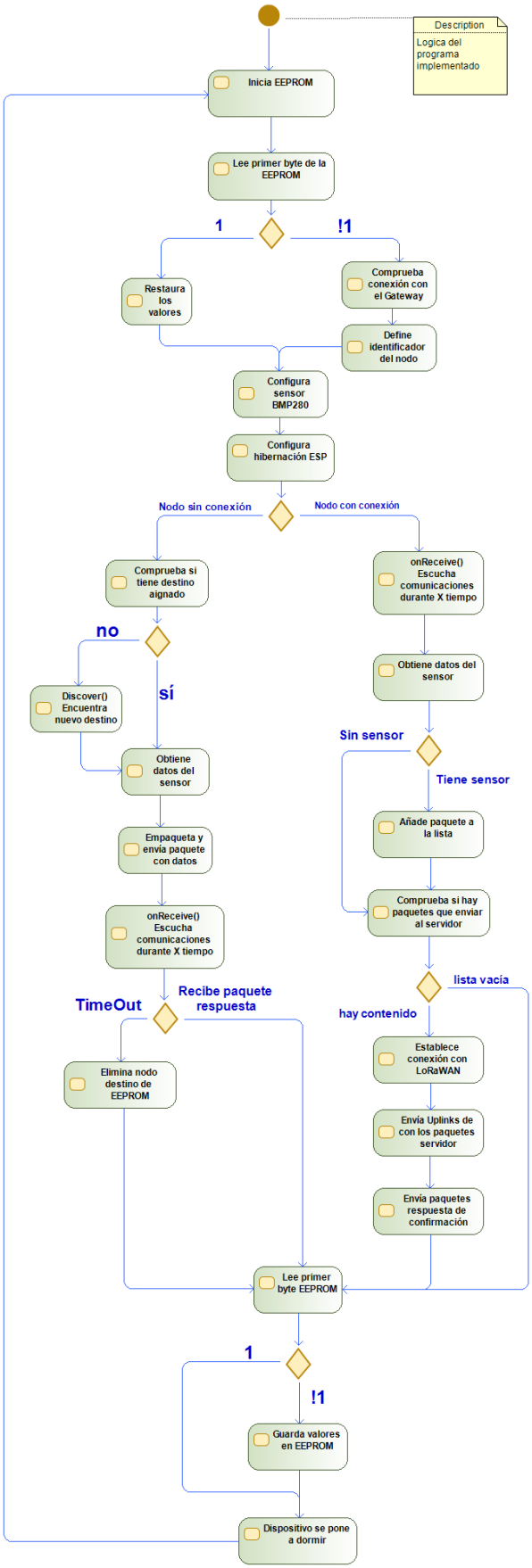

Figura 30. Diagrama de actividad del programa principal

5. VALIDACIÓN

Una vez los dispositivos están configurados y se ha implementado correctamente el programa en los nodos que componen la red LoRaWan y red LoRa mallada, hay que comprobar el funcionamiento de la solución implementada. Para la realización de pruebas se han utilizado dos nodos, un Gateway y el servidor de ChirpStack.

En esta implementación, los dispositivos realizan un envío por minuto y el contenido de estos paquetes aportan una gran variedad de información sobre la transmisión, además del payload del paquete (Ver Figura 31).

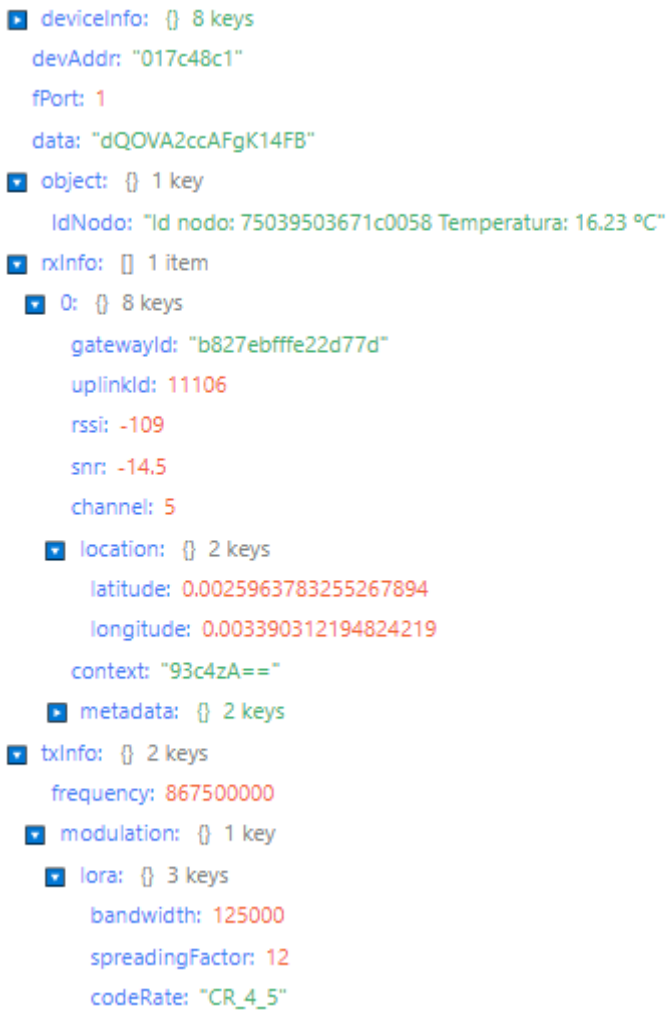

Figura 31. Contenido de los paquetes

Se ha realizado un despliegue de los dispositivos a diferentes distancias entre ellos y el Gateway para visualizar los resultados obtenidos e ir comparándolos con las distintas distancias. Las transmisiones realizadas tienen la configuración que se muestra en la Figura 32.

Figura 32. Parámetros de transmisión

5.1. PRUEBA 1: COMUNICACIÓN LORAWAN

La primera prueba consiste en comprobar la correcta comunicación de los nodos en la red LoRaWAN. Para esta prueba, los nodos están colocados junto al Gateway y ambos están activados en el servidor ChirpStack. Estos dos dispositivos van a realizar envíos a la red LoRaWAN en el orden de eventos mostrado en la Figura 33. Primeramente, los dispositivos comprueban si tiene conexión al Gateway, de ahí el primer join registrado. Después cierran comunicaciones con LoRaWAN y escuchan las de LoRa durante un tiempo. Pasado este tiempo se vuelve a conectar a LoRaWAN y envía el paquete que ha registrado de su sensor.

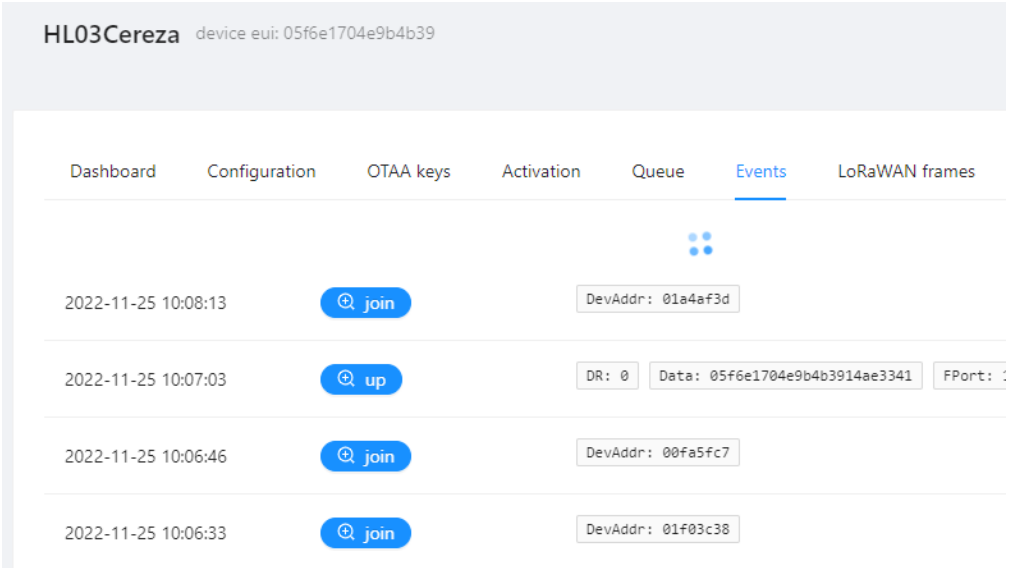

Figura 33. Recepción de paquetes en Servidor

A continuación, se muestran los resultados de la transmisión de los nodos sensores HL03Cereza y HL04Datil, recogidos por el servidor de ChirpStack, con resultados muy similares y sin pérdidas de datos.

Para la Figura 34 tenemos la información de la transmisión de 5 paquetes transmitidos por el dispositivo HL03Cereza al servidor LoRaWAN.

RSSI	SNR	FRECUENCIA	SF	BAND WIDTH
-18	7.5	868.3	12	125
-29	7.2	868.3	12	125
-29	6.8	867.9	12	125
-47	5.2	867.5	12	125
-43	6.2	867.7	12	125

Figura 34. Transmisión entre dispositivos HL03Cereza-Gateway en LoRaWAN (Prueba 1)

En la Figura 35 se muestra los datos de la transmisión de 5 paquetes desde el dispositivo HL04Datil al servidor LoRaWAN.

RSSI	SNR	FRECUENCIA	SF	BAND WIDTH
-31	5.5	868.3	12	125
-41	4.8	868.3	12	125
-37	6	867.3	12	125
-37	7	867.7	12	125
-35	5.9	867.9	12	125

Figura 35. Transmisión entre dispositivos HL04Datil-Gateway en LoRAWAN (Prueba 1)

5.2. PRUEBA 2: RED LORA MALLADA Y LORAWAN

En esta segunda prueba se pretende comprobar el correcto funcionamiento de la red mallada LoRa con la arquitectura LoRaWAN, probando que un nodo sin conexión con el Gateway es capaz de encontrar un nodo destino a través del cual entregar paquetes al servidor red de LoRaWAN.

Se realizan diferentes pruebas para descubrir en qué zonas dispone de conexión el nodo con el Gateway, moviéndolo hasta encontrar una zona sin cobertura. Se coloca pues uno de los nodos fuera del rango de conexión del Gateway, y el segundo nodo en una zona con conexión. Se espera que el nodo sin conexión, al comprobar que no está en rango del Gateway, se ponga a descubrir nodos que tengan conexión y logre establecer comunicación con el segundo nodo. El primer nodo le redirige el tráfico de sus paquetes y podemos observar desde el servidor ChirpStack, como el segundo nodo se conecta a la red LoRaWAN y envía los paquetes del primer nodo y los suyos, que contienen sus identificadores y la temperatura captada por cada uno (Ver Figura 36).

Figura 36. Recepción de datos en ChirpStack (Prueba 2)

La realización de esta prueba ha tenido lugar en un entorno urbano, donde los nodos están a una distancia de 100 metros entre sí y la distancia del Gateway entre el nodo con conexión y el nodo sin conexión es de 150 metros y 400 metros respectivamente. Este entorno urbano, debido a los obstáculos (principalmente edificios), presenta un radio de transmisión mucho menor que en un espacio abierto, sin obstáculos.

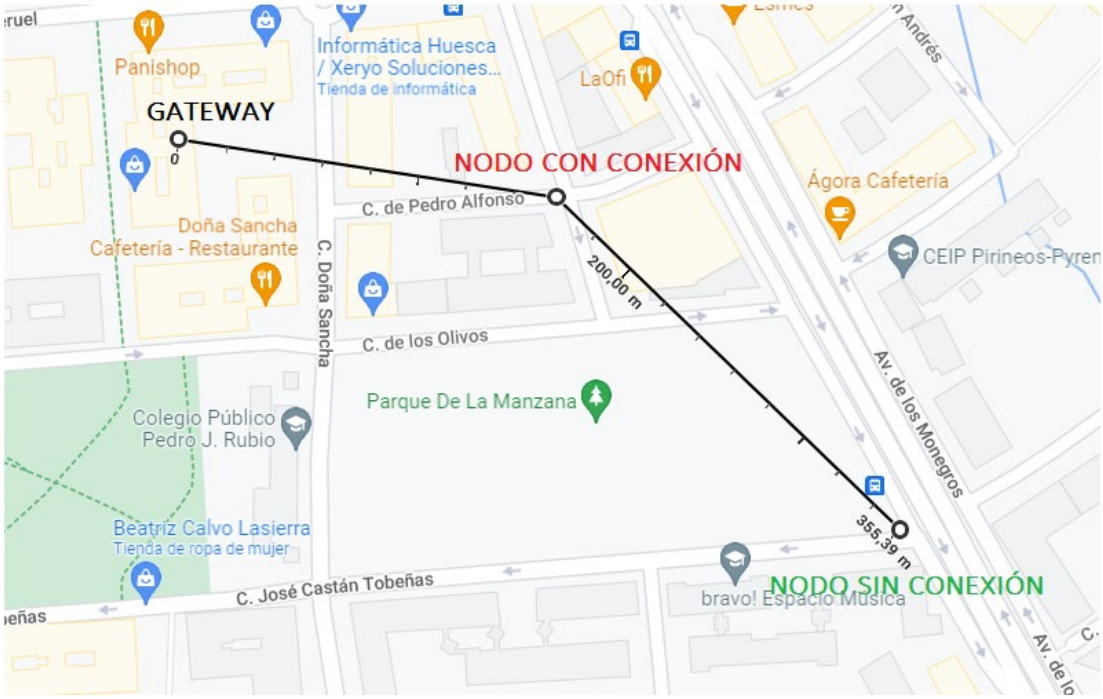

Figura 37. Distancias entre el Gateway y los nodos

Observando los paquetes recibidos en el servidor, podemos confirmar que el payload recibido es la información que ha captado cada dispositivo, pues incluye el identificador y la temperatura (Ver Figura 38 y Figura 39).

Figura 38. Payload del nodo HL04Datil (Prueba 2)

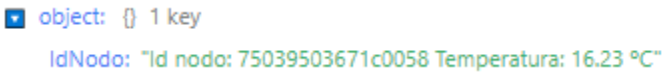

Figura 39. Payload del nodo HL03Cereza (Prueba 2)

Mediante el nodo con conexión se miden las señales RSSI y SNR de las transmisiones LoRa del dispositivo sin conexión, información mostrada en la Figura 40.

RSSI	SNR	SF	BAND WIDTH	CODE RATE
-118	12	12	125	4_5
-118	12	12	125	4_5
-119	6	12	125	4_5
-75	12	12	125	4_5

Figura 40. Transmisión LoRa entre nodos HL04Datil – HL03Cereza

En la Figura 41 se observa los niveles RSSI y SNR de los paquetes entregados por el nodo con conexión al Gateway. Entrega por cada conexión al servidor dos paquetes, el que ha recibido por parte del nodo sin conexión cuando escuchaba las comunicaciones LoRa, y el suyo propio.

RSSI	SNR	FRECUENCIA	SF	BAND WIDTH
-111	-11.8	868.3	12	125
-109	-14.5	867.5	12	125
-109	-9.2	868.5	12	125
-113	-9.5	868.1	12	125
-107	-8.8	867.7	12	125
-107	-6.2	867.5	12	125
-105	-5	867.3	12	125
-107	-7	867.7	12	125

Figura 41. Transmisión LoRaWAN entre HL03Cereza – Gateway (Prueba 2)

En estas tablas destaca la variación de RSSl y SNR, ya que, al haberse realizado estas pruebas en un entorno urbano, las interferencias influyen mucho en estas medidas y no son las que se presentan en campo abierto, que es para lo que están orientadas en aplicaciones de Agricultura Inteligente.

6. CONCLUSIONES

En este trabajo se ha realizado un estudio sobre las diferentes tecnologías LPWAN y se ha entrado en mayor profundidad en el estándar LoRaWAN, en la tecnología LoRa y su uso en redes malladas. Gracias a esto, se ha podido obtener un enfoque global sobre las tecnologías implicadas en el desarrollo de IoT y ser conscientes del gran futuro que este campo tiene debido a las millones de implementaciones diferentes que hay en el mundo real. Tan solo con la tecnología LoRa se ha visto la gran cantidad de ámbitos que se pueden abarcar, además, con la posibilidad de combinar diferentes tecnologías es posible aumentar todavía más las funciones que se pueden aportar.

Se ha demostrado con este estudio, que el diseño e implementación de la red LoRa mallada de un salto, mejora la entrega de paquetes con respecto a configuraciones más sencillas, como lo es la topología en estrella utilizada en LoRaWAN. Gracias a los protocolos de descubrimiento de nodos con conexión y gestión de comunicaciones LoRa implementados sobre la red mallada, se ha mejorado el estándar LoRaWAN, cubriendo las debilidades de esta arquitectura, haciéndola más persistente a pérdida de paquetes, mejorando las distancias de transmisión de datos y aumentando las zonas de cobertura y la cantidad de información recogida sin despliegues de Gateways adicionales.

El diseño de la red LoRa mallada es una parte fundamental a la hora de enfocar el proyecto al que va dirigido. En este trabajo, se ha decidido simplificar la red mallada a un salto, pero multitud de proyectos tienen enfoques que requieren soluciones multisalto, aumentando el número de entornos en los que se puede utilizar la tecnología LoRa.

Para una implementación real del sistema en un ámbito comercial, requeriría ampliar las funcionalidades mediante tecnologías externas para ofrecer un servicio más completo. Se debería implementar de infraestructura y plataformas que añadan funcionalidades, un sistema de almacenamiento mediante el uso de bases de datos, gestión y análisis de datos recogidos para la realización de estudios y obtención de resultados mediante el uso de Big Data y Machine Learning. Además, este proyecto puede dar pie a un futuro desarrollo de redes LoRa malladas multisalto.

REFERENCIAS BIBLIOGRÁFICAS

[1] Semtech LoRa applications for smart agriculture, Semtech, 2022.

https://www.semtech.com/lora/lora-applications/smart-agriculture

[2] Poultry farms use IoT solution for real-time supply chain insights

Lora-alliance, November 2021.

https://resources.lora-alliance.org/use-case/poultry-farms-use-iot-solution-for-real-time-supply-chain-insights

[3] Semtech solution for Cloud-Managed Asset and Facility Monitoring, Semtech, Oct. 29, 2020.

https://www.semtech.com/company/press/semtech-supports-new-iot-solutions-for-cloud-managed-asset-and-facility-monitoring-developed-by-cisco

[4] Semtech solution for remote and automatic monitoring of gas consumption, Semtech, July 30, 2020.

https://www.semtech.com/company/press/semtech-and-aiut-optimize-petroleum-gas-management-with-lora-devices

[5] Poultry farms use IoT solution for real-time supply chain insights

LoRa Alliance, November 2021.

https://resources.lora-alliance.org/use-case/poultry-farms-use-iot-solution-for-real-time-supply-chain-insights

[6] Real-time monitoring of environmental and health parameters, LoRa Alliance, December 2014.

https://lora-alliance.org/wp-content/uploads/2020/12/THE-FARMING-OF-TOMORROW-IS-ALREADY-HERE-HOW-LoRaWAN%C2%AE-TECHNOLOGY-SUPPORTS-SMART-AGRICULTURE-PRECISE-ANIMAL-PRODUCTION.pdf

[7] LoRaWAN solution for Smart farming, Mokolora, January 2021.

https://www.mokolora.com/es/solutions/smart-farming/

[8] Smart parking solution with Semtech LoRa technology, Pni sensor corporation, January 3, 2017.

https://www.pnicorp.com/wp-content/uploads/PNI-Sensor-Corp_PlacePod_Press-Release_Jan-3-2017.pdf

[9] Smart power management, Semtech, April 2017.

https://info.semtech.com/hubfs/Smart%20Cities%20Ebook/USECASE_SEMTECH_SMARTCITIES_BOOK.pdf?hsLang=en-us

[10] JAMAC multi-hop protocol for LoRa, Juan José López Escobar, Felipe Gil-Castiñeira y Rebeca P. Díaz Redondo, December 2020.

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7730183/

[11] Monitoring of large area using LoRa Mesh network,Huang-Chen Lee; Kai-Hsiang Ke, March 2018 .

https://ieeexplore.ieee.org/abstract/document/8326735

[12] Semtech boost urban forest solution, Semtech, November 2021.

https://www.semtech.com/company/press/semtechs-lora-devices-and-the-lorawan-standard-boost-urban-forest-management

[13] Semtech flood monitor solution, Semtech, October 2018.

https://www.semtech.com/company/press/semtechs-lora-technology-and-senets-lorawan-based-network-leveraged-in-flood-sensors-to-monitor-water-levels

[14] Forest Fire Detection using LoRaWireless Mesh Topology,Adnan, A.; Salam, E.U.; Arifin, A.; Rizal, 2018.

http://103.195.142.203/storage/dokumen/artikel-1619109355-Forest%20Fire%20Dtection%20Using.pdf

[15] LoRa Mesh for wide area animal tracking,Jithu G. Panicker; Mohamed Azman; Rajiv Kashyap, 2019.

https://ieeexplore.ieee.org/document/8868958

[16] LPWAN tecnology, Allionusa, October 2021.

https://www.allionusa-sgs.com/iot-lpwan/

[17] Semtech Technology overview, Semtech, 2022.

https://www.semtech.com/

[18] LoRa Frecuency, Mokosmart, June 2017.

https://www.mokosmart.com/es/lora-frequency/

[19] ETSI overview, European Telecommunications Standards Institute, 2022.

https://www.etsi.org/

[20] LoRa ISM frecuency, Pdacontroles, November 2016.

http://pdacontroles.com/introduccion-lora-modulo-rfm95-hoperf

[21] Spreading factor information, Techplayon, April 2017.

https://www.techplayon.com/lora-link-budget-sensitivity-calculations-example-explained/

[22] LoRa technology in the present, 2cigroup, February 2019.

https://www.2cigroup.com/es/conceptos-de-actualidad-lora-y-lorawan/

[23] LoRaWAN overview, Medium, August 2012.

https://medium.com/pruebas-de-laboratorio-de-la-modulaci%C3%B3n-lora/lorawan-d00f48384160

[24] LoRaWAN different nodes and classes, M2mlogitek, July 2014.
https://www.m2mlogitek.com/empezando-a-trabajar-con-lorawan-ii/

[25] Heltec LoRa node specs, Heltec, April 2018.
https://heltec.org/project/wifi-lora-32/

[26] Sensor Bmp280 description, Prometec, April 2020.
https://www.prometec.net/sensor-de-presion-y-temperatura-bmp280/

[27] ChirpStack Overview, ChirpStack open source LoRaWAN Networkk Server, 2022.
https://www.chirpstack.io/

[28] The Things Network Overview, The Things Network, 2022
https://www.thethingsnetwork.org/

[29] Amazon Web Services for LoRaWAN, Amazon, 2022.
https://aws.amazon.com/es/iot-core/lorawan/

[30] Arduino configuration for Heltec´s nodes, Heltec, September 2022.
https://docs.heltec.org/en/node/esp32/quick_start.html

[31] Heltec libraries for Esp32 and LoRa devices, Arduino, March 2014.
https://www.arduinolibraries.info/libraries/heltec-esp32-dev-boards

[32] LoRaWAN library, ElectronicCats, 2018.
https://github.com/ElectronicCats/Beelan-LoRaWAN

[33] Bmp280 sensor libraries, Adafruit, 2018.
https://github.com/adafruit/Adafruit_BMP280_Library

[34] Pin map for Heltec wifi lora 32, The Hiveeyes community, February 2017.
https://community.hiveeyes.org/t/heltec-wifi-lora-32/3125

[35] iNiT overview, Intelligent Networks and Information Technologies, 2022.
http://init.unizar.es/