UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DISCA
DEPARTAMENTO DE INFORMÁTICA DE SISTEMAS Y COMPUTADORES

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

MÁSTER UNIVERSITARIO EN INGENIERÍA DE COMPUTADORES Y REDES

DEPARTAMENTO DE INFORMÁTICA DE SISTEMAS Y COMPUTADORES

TRABAJO FIN DE MÁSTER

Sumidero híbrido para redes inalámbricas de sensores

ANTONIO RISUEÑO SÁNCHEZ

Septiembre de 2018

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

MÁSTER UNIVERSITARIO EN INGENIERÍA DE COMPUTADORES Y REDES

DEPARTAMENTO DE INFORMÁTICA DE SISTEMAS Y COMPUTADORES

TRABAJO FIN DE MÁSTER

Sumidero híbrido para redes inalámbricas de sensores

Autor: Antonio Risueño Sánchez

Directores: Dr. D. José Carlos Campelo Rivadulla

Dr. D. Alberto Miguel Bonastre Pina

Septiembre de 2018

Antonio Risueño Sánchez

Valencia – Spain

E-mail: anrisan2@posgrado.upv.es
Teléfono: +34 661 162 767

Declaración de Autoría

Yo, Antonio Risueño Sánchez con DNI 48154325F, declaro que soy el único autor del Trabajo Fin de Máster titulado **Sumidero híbrido para redes inalámbricas de sensores** y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual y que todo el material no original contenido en dicho trabajo está debidamente atribuido a sus legítimos autores.

Valencia, a 7 de septiembre de 2018

Fdo.: Antonio Risueño Sánchez

Resumen

Actualmente los dispositivos que dan vida al Internet de las Cosas son aquellos sistemas destinados a conectarse con un servicio externo al que enviar o recibir un conjunto de información u órdenes que varía en función del marco de la aplicación y de sus requisitos. La constante evolución de estos dispositivos viene determinada por varios factores. En primer lugar, el coste por dispositivo se abarata debido a la miniaturización de los componentes electrónicos. El consumo eléctrico de las baterías también es menor gracias al empleo de tecnologías de comunicación más eficientes y algoritmos que reducen el tiempo empleado para comunicarse por radio. Por último, la capacidad de integración actual permite equipar a estos dispositivos con una potencia de cálculo muy superior a la de generaciones anteriores. Estos nuevos sistemas pueden procesar en el propio chip la información obtenida con un alto grado de detalle. De esta manera, solo se envían al exterior los datos procesados que estén listos para utilizar o enviar un conjunto de datos de una sola vez. Todos estos avances permiten el uso masivo de estos dispositivos en nuevos sectores y que las aplicaciones actuales y futuras se enriquezcan en gran medida, beneficiando a usuarios, empresas, industrias, gobiernos o incluso universidades.

A medida que aumentan los escenarios posibles en el que estos dispositivos deben de conectarse y comunicarse con otras redes se definen unos nuevos retos de conectividad. En este trabajo desarrollaremos un nuevo tipo de sumidero híbrido que dará soporte a la comunicación de distintos dispositivos que forman parte de la infraestructura del Internet de las Cosas. Los sumideros híbridos permitirán a estos pequeños sistemas enviar la información que recolectan sus sensores hacia un servicio externo. En concreto desarrollaremos un diseño flexible capaz de aceptar diferentes tipos de tecnologías inalámbricas y protocolos. Los sumideros híbridos podrán ser reconfigurados en función de los cambios de la topología. El diseño podrá ser desplegado en todo tipo de entornos, ya que no requiere una posición fija de los sistemas ni los sensores. Esto a su vez posibilitará la utilización de un gran abanico de sensores y actuadores. Discutiremos las plataformas hardware de bajo coste y software de código abierto que se emplearán para implementar el sistema final y además seremos capaces de proporcionar tolerancia a fallos en ciertos puntos de la red para conseguir una infraestructura más robusta.

Una vez que el sistema este totalmente montado y sea funcional evaluaremos el diseño planteado elaborando un estudio de prestaciones sobre un entorno de pruebas comparable al mundo real. En las pruebas un dispositivo embebido enviará la información recolectada de sus sensores a una plataforma del Internet de las Cosas en la nube bajo diferentes circunstancias de conexión. Durante la ejecución de las pruebas monitorizaremos varios parámetros del diseño que nos permitirá conocer los beneficios obtenidos en este trabajo y descubrir nuevas líneas de investigación para continuar con el desarrollo de este trabajo.

Abstract

These days the devices that give life to the Internet of Things are those systems designed to connect with an external service that sends or receives a set of information or orders that varies depending on the framework of the application and its requirements. The evolution of these devices is continuing due to several factors. In the first place, the cost per device is becoming cheaper due to the miniaturization of the electronic components used. The electric consumption of the batteries is also lower thanks to the use of more efficient communication technologies and which are equipped with algorithms that reduce the time expend in the communication over the radio. Finally, the current integration capacity allows equipping these devices with more computing power than the previous generations. These new systems can process inside the chip the information obtained with a high degree of detail in the system itself. In this way, only the data that are sent abroad are ready to use or a set of data can be send at once. All these advances allow the massive use of these devices in new sectors. Furthermore current and future applications are greatly enriched, benefiting users, companies, industries, governments or even universities.

As the possible scenarios in which these devices should connect and communicate with other networks are becoming larger, new connectivity challenges are defined. In this work we will develop a network based on a new type of hybrid sink that will support the communication of the devices that are part of the Internet of Things infrastructure. The hybrid sinks will allow these small systems to send the information collected by their sensors to an external service. We will develop a flexible design capable of accepting different types of wireless technologies. The hybrid sinks can be reconfigured based on changes in the topology. This design can be deployed in all types of environments because neither the systems nor sensors need a fixed position in the field. In return this will enable the use of a wide range of sensors and actuators. We will discuss the low-cost hardware platforms and open source software that will be used to implement the final system and we will also be able to provide fault tolerance at certain points in the network to achieve a more robust infrastructure.

Once the system is fully assembled and functional, we will evaluate the proposed design, preparing a study of benefits on a testing environment that is comparable to the real world. In the tests an embedded device will send the information collected from its sensors to an-Internet of Things platform in the cloud under different connection circumstances. During the execution of the tests we will monitor several parameters of the design that will allow us to know the benefits obtained in this work and discover new lines of research to continue with the development of this work.

Agradecimientos

Quiero agradecer principalmente este trabajo a mis padres, quienes me han apoyado durante este duro camino y han hecho posible que pueda lograr cada una de mis metas. También al resto de mi familia y amigos de toda la vida que siempre han confiado en mí y me han dado la fuerza para superar cualquier obstáculo.

A mis directores del trabajo de fin de máster, José Carlos por sus buenos consejos y su valioso asesoramiento durante toda la realización de este proyecto. Y Alberto, quien ha contribuido con algunas de las partes más esenciales de este trabajo y ha supuesto una fuente de inspiración durante todo el desarrollo.

A las personas que forman parte de la comunidad del software libre, cuyo trabajo ha ayudado a enriquecer la elaboración de estas páginas.

Dedicatorias

Dedico estas líneas y todos los logros que he conseguido durante mi vida a mis padres, que una vez más han permitido que continúe con mis estudios, y a mi hermana, por seguir demostrándome que todo es posible si trabajas para conseguirlo.

A todos mis compañeros del máster. Gracias a ellos hemos podido sacar adelante el máster y vivir un año cargado de experiencias fuera de casa.

A todos los profesores del Departamento de Informática de Sistemas y Computadores por invertir en mí su tiempo y sabiduría para que este proyecto salga adelante, y especialmente a Antonio Robles por ser el director que este máster se merece.

Índice de Contenido

Índice de Figuras

Índice de Tablas

CAPÍTULO 1. INTRODUCCIÓN

El Internet de la Cosas es el concepto que ha surgido de la idea de conectar el mundo físico a Internet. Esta idea está presente desde los años 90, pero es ahora cuando realmente está ganando importancia y cada día más y más aplicaciones se suman a este fenómeno que está revolucionando muchos de los sistemas y servicios que usamos cada día. Actualmente tenemos conectados a la red muchos dispositivos como ordenadores, Smartphones, tablets, … pero más de 50 mil millones de nuevos dispositivos se conectarán a la red en 2020 [1]. Los electrodomésticos y otros aparatos de nuestra casa que tradicionalmente operaban por sí solos con una función muy específica ahora se conectan a la red para ofrecer más características y funciones. Este fenómeno es palpable a través de varios ejemplos que ya están presentes en nuestras vidas como, por ejemplo, los televisores inteligentes que ahora pueden acceder a diversas plataformas de vídeo bajo demanda y proporcionar contenido más allá de la televisión en vivo, nuestro termostato es capaz de ajustarse automáticamente basándose en nuestros hábitos, el frigorífico puede conectarse con el supermercado y comprar los productos que se nos han acabado o que están a punto de hacerlo, incluso las bombillas ahora están conectadas a la red y nos permiten elegir la intensidad apropiada en cada momento, ser controladas remotamente o automatizar todo nuestro hogar añadiendo también otros mecanismos como las persianas o el climatizador. También observamos esta tendencia fuera del hogar: en las ciudades con la construcción de parkings inteligentes, en las industrias con la inclusión de máquinas autónomas, la creación de nuevos sectores como la agricultura de precisión, sistemas de seguridad avanzados y muchas más aplicaciones.

Con todas estas nuevas cosas conectadas el IoT nos brinda un mundo mucho más conveniente, eficiente y que nos proporciona un nuevo nivel al cual podemos obtener información sobre los usuarios. Estos datos más precisos sobre las personas que interactúan con los servicios de IoT genera un nuevo universo de oportunidades a desarrolladores y empresas que podrán enfocar sus productos a los usuarios y satisfacer con éxito las necesidades de éstos. A estas alturas, la mayoría de las empresas tecnológicas tienen en su cartera proyectos de IoT listos para entrar en funcionamiento ampliando así sus oportunidades de negocio y sus beneficios. En este trabajo se busca proponer una filosofía de redes múltiples con encaminamiento adaptativo que posibilitará la creación de nuevas aplicaciones empleando hardware de bajo coste y un servicio en la nube. Este diseño permite utilizar cualquier tecnología y protocolo dentro de la red para comunicarnos con el resto de los dispositivos.

1.1. MOTIVACIÓN

Como acabamos de ver la revolución del Internet de las Cosas no ha hecho más que comenzar y todavía existen muchos campos en los que investigar para garantizar el éxito de esta tecnología. En nuestro caso queremos mejorar la red que montan los dispositivos de IoT, ya que la mayoría son inalámbricas. Esto implica que la red siempre está sujeta a interferencias, desconexiones y otros fenómenos que pueden provocar inestabilidades. La realización de este trabajo permitirá progresar en el campo de la conectividad para aumentar la fiabilidad y confiabilidad de las comunicaciones dentro de los sistemas IoT. La meta final de este trabajo es diseñar un sumidero que sigue un nuevo paradigma, conseguir que un nodo pueda emplear cualquier tecnología que tenga alcance para llevar la información desde el resto de los nodos hasta el exterior.

Este trabajo propone la creación de un diseño que reúne los requisitos de conectividad y disponibilidad para las aplicaciones de IoT actuales. Con la implantación de este diseño y su posterior evaluación en un entorno de pruebas se podrá analizar el funcionamiento del sistema durante su ejecución, la interacción entre todos los dispositivos que forman la red, su grado de disponibilidad, su comportamiento frente a fallos y los mecanismos de seguridad integrados. Una vez se concluya este trabajo se extraerán las ventajas y desventajas del diseño y se valorará qué aspectos se pueden mejorar en futuras versiones. Los resultados obtenidos en este trabajo nos ayudarán a determinar qué sistemas y tecnologías son válidas para este diseño y cuáles pueden ser consideradas para su implantación en desarrollos futuros a medida que estas evolucionan.

1.2. OBJETIVOS

El objetivo principal de este trabajo es aumentar la confiablidad de los sistemas de IoT mediante la creación de una red de dispositivos recolectores capaces de adaptarse a fallos en los enlaces de comunicación hacia las plataformas de IoT. Los sumideros híbridos permitirán dar soporte de conectividad a múltiples tipos de dispositivos, trabajar con varios protocolos y combinar diferentes tecnologías inalámbricas para establecer rutas alternativas hacia Internet. A continuación se detallan algunos objetivos secundarios y específicos del trabajo.

1.2.1. OBJETIVOS SECUNDARIOS

Los objetivos principales de este trabajo están numerados a continuación:

a) Estudiar las características de algunas plataformas de IoT actuales, sus funcionalidades y qué nos ofrecen para desarrollar nuevas aplicaciones.

b) Estudiar un algoritmo de encaminamiento que nos permita enrutar tráfico dentro de una red en malla y desplegarlo en nuestro prototipo.

c) Seleccionar los dispositivos más apropiados para desarrollar nuestra idea de sumidero híbrido y analizar su desempeño en el prototipo.

d) Comprobar a través de una batería de pruebas la funcionalidad del prototipo propuesto y su viabilidad como punto de partida para trabajos futuros.

e) Establecer una vía de diseño para que el prototipo propuesto sea escalable.

1.2.2. OBJETIVOS ESPECÍFICOS

Además de los objetivos principales en este trabajo desarrollaremos los siguientes objetivos específicos para ayudarnos a alcanzar el objetivo principal:

a) Poner en marcha un modelo de red que acepte un conjunto de tecnologías y protocolos reducido como primer paso para llegar a un diseño más complejo.

b) Proporcionar tolerancia a fallos y reconfiguración automática de la red en caso de necesidad o como resultado de la toma de decisiones del nodo híbrido.

c) Crear una red en malla empleando diferentes dispositivos inalámbricos y probar su viabilidad para el diseño planteado.

d) Crear una batería de pruebas que nos permita observar el comportamiento del sistema y la plataforma IoT bajo diferentes entornos de funcionamiento.

e) Conocer y configurar las placas Raspberry Pi 3 y BeagleBoard para implementar el prototipo propuesto.

f) Investigar el ecosistema de dispositivos que nos proporciona el fabricante Pycom para integrar diferentes tecnologías y protocolos en el diseño.

1.3. ESTRUCTURA DE LA MEMORIA

El resto del documento se divide en cuatro capítulos, el capítulo dos comprende todo el estado del arte del Internet de las Cosas. En este capítulo se describen las tecnologías que hacen posible las plataformas del Internet de las Cosas, su diferenciación con la computación en la nube y se expone una clasificación de plataformas de IoT según sus características. Más tarde se detallan varios tipos de redes de sensores inalámbricas mostrando sus principales componentes y funcionalidades. Al final del capítulo hablaremos de dos plataformas de IoT que consideramos de relevancia para el desarrollo de este trabajo.

En el capítulo tres se desarrolla el diseño propuesto para la plataforma IoT. Primero se realiza un análisis introductorio al concepto de red que persigue este trabajo, una red de sensores híbrida. Posteriormente hablaremos del hardware empleado para conseguir un diseño base derivado de la idea principal. A continuación, nos detendremos en el software y las plataformas de IoT analizadas, las herramientas empleadas para su implementación y la configuración utilizada en los sensores. En el último punto de este capítulo se describirá como interactúa el sistema en conjunto y cuál será el flujo de los datos.

En el capítulo cuatro probaremos el prototipo propuesto en entornos de pruebas y se realizará una evaluación de la infraestructura diseñada. Describiremos las pruebas a realizar y cómo han sido concebidas y adaptadas a este trabajo. Compararemos las funciones que nos ofrece nuestro prototipo con los objetivos del trabajo, con el fin de garantizar la viabilidad del sistema implementado para su reutilización en trabajos posteriores.

Por último se encuentra el capítulo cinco, las conclusiones. Durante este capítulo discutiremos los avances conseguidos con la realización del prototipo planteado. Si éste cumple con los objetivos propuestos en este trabajo y qué es lo que se ha aprendido.

CAPÍTULO 2. INTERNET DE LAS COSAS

En este capítulo describiremos de qué se compone el Internet de las Cosas, sus características y ventajas principales. Además se dará un ejemplo de uso para ilustrar la funcionalidad del Internet de las Cosas relacionándola de manera práctica con una aplicación en la vida real. Por último hablaremos de las redes que dan vida a las redes de sensores del futuro y mostraremos una serie de plataformas en línea que ya cuentan con servicios específicamente diseñados para estos dispositivos y adaptados a sus necesidades. Gracias a estas plataformas podremos sacar el máximo provecho a esta tecnología.

2.1. FACTORES DE ÉXITO DEL IOT

En primer lugar enumeraremos los principales factores que favorecen el crecimiento del ecosistema IoT y su implantación en cada vez más sistemas. A partir de aquí podremos entrar en más detalle en la infraestructura del IoT. Los factores son los siguientes [2]:

1. **Hardware de bajo coste**: El coste de los dispositivos (sensores, actuadores, dispositivos de interconexión, ...) con los cuales podemos diseñar e implementar un sistema de IoT se está reduciendo en gran medida. En el mercado tenemos a nuestra disposición un gran surtido de marcas que desarrollan y fabrican sus dispositivos orientados a aplicaciones de IoT, donde el coste por sistema es relativamente bajo, ya que son necesarios varios de ellos y pueden centrar su margen de beneficio en las plataformas en línea que acompañan a esos dispositivos.

2. **Hardware pequeño pero con altas prestaciones**: La huella de los dispositivos se ha estado reduciendo con el paso de los años. Los microcontroladores que antes podían ocupar unos centímetros en una placa ahora ocupan milímetros e incluso nanómetros con la escala de integración actual. Esto lleva consigo una serie de beneficios como la reducción del consumo eléctrico, la integración de estos dispositivos en más entornos sin que sea perceptible su instalación, mayor número de tecnologías inalámbricas integradas en un mismo circuito, lo que a su vez nos permite comunicar ese dispositivo con más redes y de manera más fiable.

3. **Movilidad asequible**: Las tarifas de datos móviles que antes suponían un impedimento para muchas aplicaciones de IoT, por su gran coste, ahora son mucho más baratas, aceptan más tipos de aplicaciones como el VoIP e incluyen más tráfico de datos y más ancho de banda con el uso de nuevas tecnologías como el 4G [5] o en un futuro con el 5G. Además existen nuevos protocolos diseñados específicamente para estos sistemas como Sigfox [3] o NB-IoT (Narrowband IoT) [4].

4. **Posibilidad de soportar diferentes herramientas**: Actualmente el conjunto de herramientas capaces de diseñar y poner en marcha una red de IoT son más numerosas que nunca. Los desarrolladores hacen que cada vez sea más fácil utilizar los dispositivos presentes de IoT con sus plataformas y se conecten a ellas de manera rápida y sencilla. Incluso existen plataformas que ya permiten analizar los datos obtenidos por sensores y otros dispositivos y enviarlos a grandes ordenadores para su proceso con sistemas de Big Data. Gracias a estos sistemas podremos mandar órdenes de vuelta que nos permitan alcanzar los objetivos de nuestra aplicación.

5. **Ya están presentes en el mercado**: Como ya adelantábamos el IoT ha venido para permanecer con nosotros y ayudarnos a interconectar cada dispositivo que poseamos con el mundo que le rodea. Sin duda está suponiendo una revolución y que un futuro lo encontraremos por todas partes.

2.2. INFRAESTRUCTURA DEL IOT

Las redes de sensores forman parte de las tecnologías esenciales que posibilitan el despliegue de IoT. Para su desarrollo son necesarios los siguientes componentes [2]:

1. **Hardware**: En primer lugar nos hacen falta los dispositivos que interactúan con el mundo real de alguna manera, como registrar un determinado fenómeno. El hardware abarca los sistemas físicos, sus microcontroladores, sensores, actuadores, baterías y hardware de comunicación. Actualmente podemos contar con hardware muy diverso pero que en su mayoría contarán con un SoC ARM debido a que nos proporcionan una gran capacidad de computo a bajo coste y con un consumo eléctrico controlado. Estos chips contienen un sistema completo dotado de CPU, memoria RAM y ROM, interfaces de entrada y de salida e incluso una GPU.

2. **Comunicación**: La comunicación permite que la información viaje desde los dispositivos hasta la plataforma que los procesa o los muestra y al contrario. Este elemento es vital para que el sistema funcione en conjunto ya que se asegurará de que los dispositivos pueden conectarse con una red, ya sea a través de protocolos propietarios o de código libre.

3. **Firmware**: El firmware es el software más básico de un sistema IoT. El firmware define el comportamiento más básico de los dispositivos, como la manera de adquirir los datos, la forma de aplicar las acciones recibidas, etc.… Por otro lado, las plataformas en línea disponen de varias herramientas básicas que se encargan de recibir la información y de proporcionar información sobre todos los dispositivos que forman parte de la red de sensores. Otras herramientas se encargan de tratar la información recibida y hacer un pos-proceso de ésta para presentarla de manera ordenada a los usuarios de esa plataforma o convertir esa información para entregarla a otro servicio, aplicación empresarial o plataforma.

4. **Aplicaciones**: Las aplicaciones es el conjunto de programas que engloban al software ya existente en los dispositivos de IoT. Las aplicaciones son capaces de utilizar las herramientas existentes para reunir distintos tipos de sensores o actuadores y a las plataformas de IoT. Todo ello con el objetivo de desarrollar una tarea concreta. También sirven como interfaz de acceso a la información que las plataformas tienen almacenada y como puerta de comunicación con otros sistemas. Las aplicaciones suelen funcionar sobre ordenadores, teléfonos inteligentes, tabletas, dispositivos con capacidades de navegador web y otras plataformas que puedan tratar esa información a través de una API.

La seguridad es uno de los componentes principales que debe de estar presente en todos los puntos anteriores. La seguridad se implementa a través de diferentes mecanismos. Estos se encargan de proteger al sistema contra ataques externos o detectar vulnerabilidades que puedan poner en riesgo la integridad de los datos que se transportan de un sitio a otro.

Un sistema fiable deberá de implementar los siguientes mecanismos en su infraestructura: cifrado de la información, gestión de los dispositivos, autentificación de usuarios, metadatos firmados y servicios anti-intrusión. Implementar todos estos sistemas puede suponer un aumento del coste de la plataforma y el tiempo necesario para poner en marcha el sistema. Sin embargo, debemos de asegurarnos que toda la información contenida y los usuarios permanezcan a salvo.

2.3. COMPONENTES DE UNA PLATAFORMA DE IOT

Las plataformas IoT son la pieza central de la arquitectura del Internet de las Cosas que trabajan en conjunto para comunicar los objetos del mundo virtual con las cosas del mundo real. Para que esto sea posible es necesario que todos los componentes de la plataforma trabajen de manera coordinada y todos los objetos puedan comunicarse entre sí. Existen 8 componentes principales en el IoT que pasaremos explicar a continuación [2]:

1. **Conectividad y normalización de los datos**: Cada plataforma de IoT comienza con una capa de conectividad que se encarga de reunir diferentes protocolos y formatos de datos en una misma interfaz software. Esta capa es necesaria para asegurarnos que todos los dispositivos pueden interactuar con los datos de manera correcta. Al tener toda la información en un mismo lugar y en un mismo formato podemos plantear el resto de las capas sobre la infraestructura de IoT, monitorizar, gestionar y analizar el sistema como conjunto.

   Esta capa puede convertirse en el elemento más complicado del sistema ya que requiere que dispositivos diferentes trabajen con un mismo formato de datos que debe poder ser utilizado e interpretado por cualquier plataforma. La forma más sencilla de proporcionar este servicio es desarrollando una API que da acceso a los datos de una manera normalizada.

2. **Gestión de dispositivos**: El módulo de gestión de dispositivos de la plataforma de IoT se encarga de que los objetos conectados están trabajando correctamente y que el software que están ejecutando funciona y está actualizado. Este módulo también se encarga del aprovisionamiento de nuevos objetos, configuración remota, actualizar el firmware y solución de problemas.

Las plataformas de IoT pueden alcanzar a tener miles de dispositivos individuales conectados a ella. Por tanto las acciones en masa y la automatización son imprescindibles para que el sistema pueda ser administrable.

3. **Base de datos**: La base de datos es la pieza central de la plataforma de IoT. La gestión de la información de los dispositivos eleva los requerimientos de la base de datos a un nuevo nivel que consiste en:

    a. **Volumen**: La cantidad de datos que se necesita almacenar puede ser gigantesca. En la mayoría de las soluciones de IoT solo se almacena una mínima parte de los datos que han sido generados, es decir, los resultados obtenidos del procesamiento de los datos de entrada.

    b. **Variedad**: Dispositivos diferentes y sensores diferentes producen tipos diferentes de datos.

    c. **Velocidad**: Muchas aplicaciones de IoT requieren que los datos que se están recibiendo sean analizados para hacer decisiones instantáneas.

    d. **Veracidad**: En algunas circunstancias los sensores pueden producir datos ambiguos o imprecisos que la plataforma debe poder detectar.

    Normalmente las plataformas de IoT vienen asociadas con una solución en la nube para el almacenamiento de datos que pueda proporcionar los requisitos que estas aplicaciones requieren. Estas soluciones puedes ser escaladas horizontal y verticalmente, pueden añadir aplicaciones de Big Data y dan servicio tanto para bases de datos relacionales como para bases de datos no relacionales.

4. **Procesado y gestión de acciones**: La información que se captura en el módulo de conectividad y normalización y se almacena en la base de datos cobra vida en este módulo donde ser procesa a través de un sistema basado en reglas o basado en la toma de decisiones a partir de un suceso específico. Estos sistemas producen una respuesta inteligente frente a un evento o la situación que detecte un sensor.

    En un sistema domótico se definen una serie de disparadores que se ejecutan cuando se detectan unas determinadas situaciones. Por ejemplo, un termostato inteligente, en base a preferencias del usuario, puede activar la calefacción o el aire acondicionado o ajustar la temperatura de alguno de los anteriores si detectamos que la temperatura de una de las habitaciones no es adecuada durante un periodo de tiempo o si el usuario abandona el hogar.

5. **Análisis de datos**: Algunas aplicaciones de IoT requieren un procesado detallado de la información para producir una respuesta o una acción. Este puede ser un proceso complejo que requiera más módulos encargados de realizar cálculos dinámicos de los sensores, sistemas de predicción y aprendizaje máquina profundo. Estos sistemas se anexan a la infraestructura del IoT básica y necesitan de otros mecanismos de comunicación para manipular datos de la base de datos y realizar acciones en el resto de los componentes.

Reutilizando el ejemplo anterior, una infraestructura de IoT que cuente con un sistema de aprendizaje puede llegar a conocer los hábitos del usuario y mantener una temperatura adecuada cuando se encuentre en casa y ahorrar energía y recursos cuando no haya nadie en casa. Incluso podemos emplear información externa al sistema, como las condiciones meteorológicas, para mejorar aún más el aprovechamiento de los recursos empelados.

6. **Visualización de los datos**: La manera en la que el usuario podrá interactuar con los datos de la plataforma de IoT será a través de gráficas analíticas que la aplicación le proporcione y será gracias a ellas las que permiten interpretar los datos recibidos fácilmente. Un buen conjunto de gráficas con los datos correctos permite a un usuario observar patrones u observar tendencias que le permitirán decidir qué hacer a continuación. Las representaciones que podemos incluir son gráficos en línea o apilados, gráficos en círculo o incluso modelos en 3D. Las gráficas que se muestran en la interfaz del panel de control del administrador también se consideran parte de esa vista analítica dentro de la plataforma de IoT.

7. **Herramientas adicionales**: Algunas plataformas de IoT también permiten la inclusión de herramientas adicionales que dan más funcionalidad al sistema. Las herramientas de desarrollo permiten al desarrollador de IoT crear prototipos y probarlos para próximas aplicaciones. Estas herramientas pueden llegar a ser muy intuitivas ya que permiten crear hasta aplicaciones de móviles con solo arrastrar los sensores a una zona de la pantalla y automáticamente aparecerán los nuevos datos.

   Existen otras herramientas que están orientadas a la administración de la plataforma de IoT y permiten obtener información sobre los dispositivos conectados, quién puede tiene acceso a la información y los usuarios actualmente conectados al sistema. Podemos descargar la información contenida en la plataforma usando herramientas de exportación de datos y obtenerla en distintos formatos estructurados que podemos utilizar en otras herramientas de análisis o guardarla a modo de copia de seguridad. Las API son un ejemplo de este tipo de herramientas ya que nos permiten obtener un conjunto en concreto de datos o filtrar información según nuestras necesidades.

8. **Interfaces externas**: Las plataformas de IoT implantadas en una empresa y que se utilizan en producción como parte de un producto no se encuentran nunca aisladas. Todas esas plataformas tienen una serie de interfaces que permiten la comunicación con servicios externos y que se utilizan para integrar esa plataforma en otro producto ya existente, como un sistema ERP, sistemas de fabricación industrial o sistemas que se encuentran dentro del ecosistema de las tecnologías de la información.

   Como ya hemos comentado, las API son una forma de acceder a la información de una plataforma de IoT, pero también se puede hacer a los recursos disponibles a través de un kit de desarrollo o SDK. Una interfaz externa bien definida puede llegar a reducir los costes de integración e implementación de una empresa pasando de meses de desarrollo a unos días.

En la Figura 1, obtenida de la referencia [2], podemos observar todos los módulos que componen la infraestructura de IoT y cómo se ordenan desde el nivel más básico hasta la inclusión de una plataforma de IoT con otro servicio.

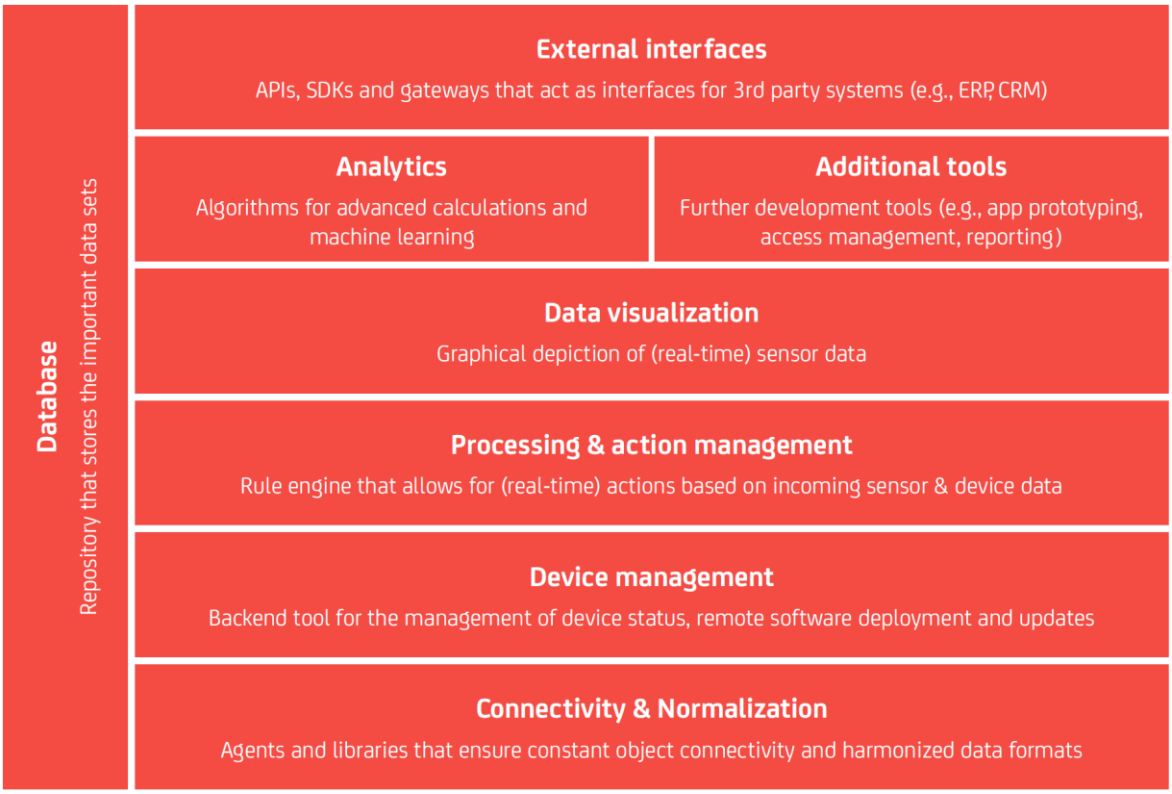


Figura 1 Componentes de las plataformas de IoT.

2.4. CASO DE USO DE IOT

A continuación vamos a presentar un caso de uso real del Internet de las Cosas para que el lector de este trabajo pueda entender de manera práctica cómo funciona un sistema de estas características. En concreto el sistema será responsable de la monitorización de las condiciones de un terreno. El usuario podrá consultar los datos desde su teléfono y podrá activar los sistemas de riego en caso necesario o programarlo para que se active de manera automática dadas unas condiciones. En la Figura 2 se ejemplifica gráficamente este ejemplo.

En este caso de uso el uso del IoT nos brinda los siguientes beneficios:

- Podemos detectar temperaturas extremas en el terreno y responder ante esas condiciones para evitar una pérdida de los cultivos que tenemos sembrados.
- El usuario será capaz de ver el estado actual de las condiciones ambientales en varios puntos del terreno y activar el sistema de riego o desactivarlo para ajustarlo a unas condiciones óptimas para la cosecha.
- El usuario podrá automatizar ciertas acciones mediante la creación de reglas.
- Los datos recogidos en esa porción de tierra pueden ser enviados a una plataforma que advierta de las próximas lluvias en base a presión atmosférica.

Gracias a estos tipos de plataformas podemos beneficiar tanto a usuarios como a empresas, ya que los datos obtenidos pueden reutilizarse para realizar distintos análisis sobre el estado del campo, determinar la calidad de la materia prima que luego se compra, establecer el precio de los intermediarios, solicitar ayudas al gobierno, etc.…

Existen cuatro elementos principales en este caso de uso:

1. **Conectar la red de sensores a una plataforma de IoT**: Este elemento es el encargado de la comunicación entre los sensores y la plataforma de IoT. Podemos emplear diferentes tecnologías inalámbricas para lograr esta conexión o un sumidero que agregue la información de un grupo de sensores básicos.
2. **Las condiciones del terreno se envían a la plataforma de IoT**: Una vez que tenemos instalados y configurados los sensores, la información que recojan se enviará a la plataforma en línea. Una vez allí los datos sobre las condiciones del terreno pueden ser consultados por el usuario en cualquier momento y a través de una interfaz accesible desde su teléfono móvil.
3. **La plataforma de IoT se encarga de accionar el sistema de riego**: El usuario será capaz de configurar los límites críticos para cada uno de los sensores de los que dispongan los dispositivos (temperatura, humedad, presión, …). Una vez sobrepasado el límite establecido la plataforma avisará al usuario y realizará la acción configurada automáticamente. El usuario siempre tendrá el control de los sistemas de riego y podrá elegir su comportamiento, aunque una regla anterior indique otra cosa.
4. **Los datos se envían a otra plataforma para obtener predicciones**: Gracias a la información recogida por los sensores podremos predecir cuándo se producirán las próximas lluvias y así optimizar el uso del agua de riego.

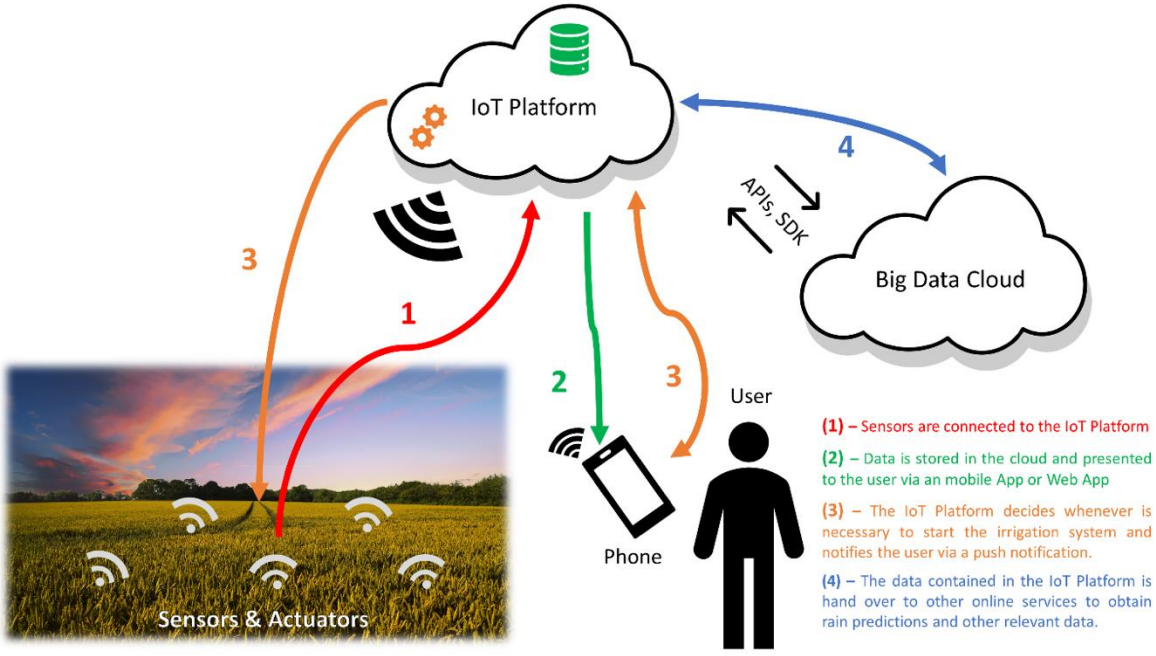


Figura 2 Caso de uso de una Plataforma de IoT.

2.5. TIPOS DE PLATAFORMAS DE IOT

En esta sección se describirán los diferentes tipos de plataformas de IoT que podemos encontrar en el mercado. Gracias a una caracterización precisa de cada una de ellas podremos escoger cual es el tipo de plataforma de IoT que podrá cubrir nuestros requisitos y necesidades con mayor acierto e invirtiendo el menor capital y tiempo posible para ponerla en marcha. Para abarcar este punto llevaremos a cabo un análisis según tres aspectos fundamentales: El nivel tecnológico de la plataforma, el mercado a cuál se orienta esa plataforma y la implementación y personalización de la plataforma [2].

NIVEL TECNOLÓGICO

Existen tres niveles tecnológicos dependiendo del número de módulos que la plataforma de IoT tenga integrados actualmente:

- **Nivel 1**: La plataforma de IoT solo realiza la recolección de datos y sirve como un bus de datos que contiene los mensajes recibidos.

- **Nivel 2**: La plataforma de IoT se encarga de la recolección de datos y bus de datos y además es capaz realizar acciones en base a las reglas que tengamos programadas. Estas plataformas ofrecen el mínimo funcionamiento para poder programar aplicaciones básicas de IoT.

- **Nivel 3**: Estas plataformas realizan las acciones de los otros dos niveles e implementan el resto de los módulos como las interfaces externas, soporte para múltiples protocolos y estándares e implementan todo el sistema de base de datos necesario además de permitir la inclusión de otros sistemas de análisis. Estas plataformas son las consideradas como las más completas ya que permiten desarrollar e implementar toda una aplicación en el mismo lugar.

En la Figura 3, obtenida de la referencia [2], tenemos una representación gráfica a modo resumen de los niveles tecnológicos que acabamos de explicar.

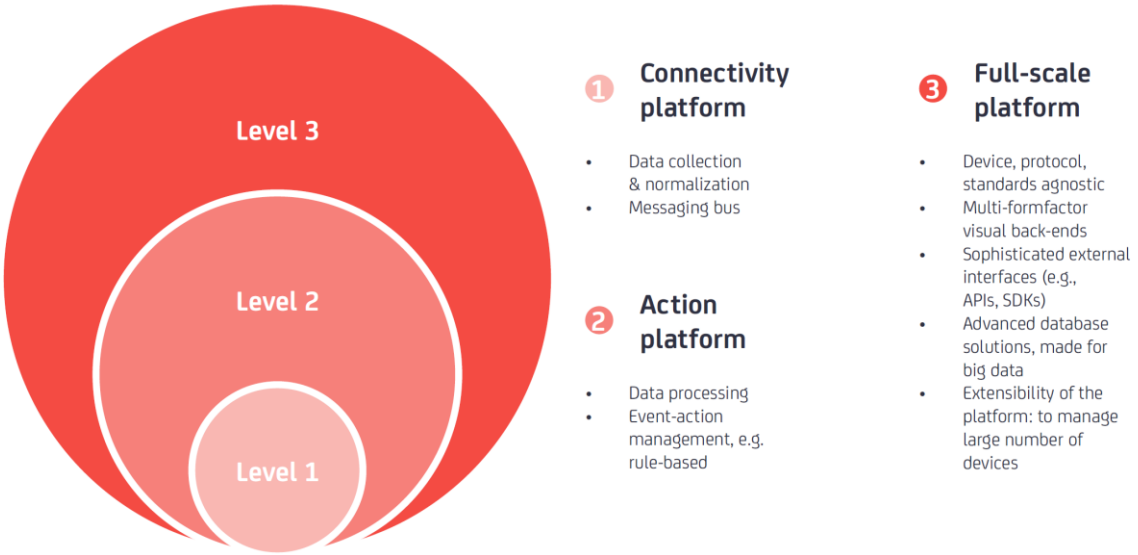


Figura 3 Niveles tecnológicos de las plataformas de IoT.

MERCADO OBJETIVO

Cada plataforma oferta diferentes servicios dependiendo de su mercado objetivo. Podemos categorizar cada plataforma de IoT según sus características, funcionalidad, protocolos que implementa o por los estándares que cumpla. Gracias a esto podemos diferenciar hasta seis tipos de mercado pero que no quedan limitados ya que en un futuro aparecerán nuevas necesidades que permitan inaugurar un nuevo sector.

- **Sector Aficionado**: Este sector está enfocado a aquellas personas que disfrutan creando sus pequeños proyectos de automatización o de Internet de las Cosas, que utilizan pequeños ordenadores como la Raspberry Pi o la placa Arduino. Las plataformas dentro de este sector tienen normalmente un coste reducido, el software que emplean es gratuito y de código abierto.

- **Hogar Inteligente**: Las plataformas domóticas soportan estándares como Wifi, Zigbee, Z-wave y Bluetooth. Vienen acompañadas de aplicaciones muy sencillas que sirven para monitorizar o controlar aparatos y dispositivos dentro del hogar.

- **Coche Conectado**: Las plataformas que permiten que los coches se conecten a la red utilizan los protocolos tradicionales de comunicación de los coches, pero también nuevos protocolos como la comunicación de coche a coche (V2V). Estas plataformas proporcionan contenido para los sistemas de infoentreteniemiento y envían determinadas métricas del coche al fabricante o a la compañía de seguros. Estas plataformas tienen muy en cuenta la seguridad de sus sistemas ya que un acceso no autorizado puede resultar fatal para el conductor del vehículo, sus ocupantes y/o el resto de los usuarios de la vía.

- **Proveedores Conectados**: Los proveedores que manejan una gran cantidad de inventario de dispositivos necesitan herramientas para poder administrar esos dispositivos y a sus usuarios fácilmente. Además estas herramientas se pueden estar integradas con otros sistemas como un CMS.

- **Ciudades Inteligentes**: Las ciudades pueden ofrecer servicios realmente útiles empleando redes de sensores o información sobre el transporte en tiempo real. Por ejemplo podemos implantar sensores de aparcamiento u ofrecer la posición real de las líneas de autobuses. Las redes empleadas en las ciudades normalmente implican el uso de redes en malla o redes de área amplia y bajo consumo (LPWAN).

- **Sector Industrial**: Las plataformas de IoT industriales proporcionan herramientas de integración con sistemas SCADA y otros automatismos. Al igual que los sistemas de los vehículos la seguridad en estas plataformas es esencial para operar de manera confiable y evitar filtraciones de información a los consumidores o la competencia.

- **Otros**: Este sector abarca otras soluciones que ya existen hoy en día pero que están en pleno desarrollo. Ejemplo de estas plataformas son: la agricultura de precisión, que busca el máximo grado de crecimiento de las plantaciones que

se realizan satisfaciendo todas las necesidades que van apareciendo en las plantas o los sistemas médicos, que son dispositivos que acompañan a los pacientes y monitorizan sus constantes continuamente, en caso de que surja alguna anormalidad se notifica al médico para que se actué inmediatamente.

NIVEL DE INTEGRACIÓN

El tercero y último diferenciador es la manera en la que las plataformas IoT pueden integrarse con la aplicación. Algunas soportan muchas posibilidades a la hora de la implementación e incluso pueden ofrecer la posibilidad de convertirse en una plataforma de marca blanca que luego personalizar con la imagen y colores de la empresa que la utilice.

- **Plataformas de todo incluido**: Estas plataformas vienen acompañadas con el soporte para un conjunto de dispositivos hardware predefinido y unas herramientas específicas para esos dispositivos. Ofrecen una solución integrada para la generación de prototipos, uso de APIs estandarizadas y apenas podemos personalizar la plataforma para ofrecerla como un producto propio. La mayoría de las plataformas que están orientadas a pequeñas empresas se incluyen en esta categoría, dejando las soluciones a media en el siguiente grupo.

- **Plataformas de marca blanca y muy personalizables**: Podemos encontrar este tipo de plataformas en el segmento de alta gama de empresa a empresa (B2B), soportan muchas opciones para la implementación y de puesta en marcha. Algunas de estas plataformas ofrecen una personalización completa de toda la interfaz a la cual podemos añadir nuestro logo, nombre y colores corporativos. Gracias a esto conseguimos desplegar un servicio en muy poco tiempo centrándonos en el producto y sus clientes y dejando las labores técnicas a la plataforma de IoT que es la encargada de proporcionar el servicio.

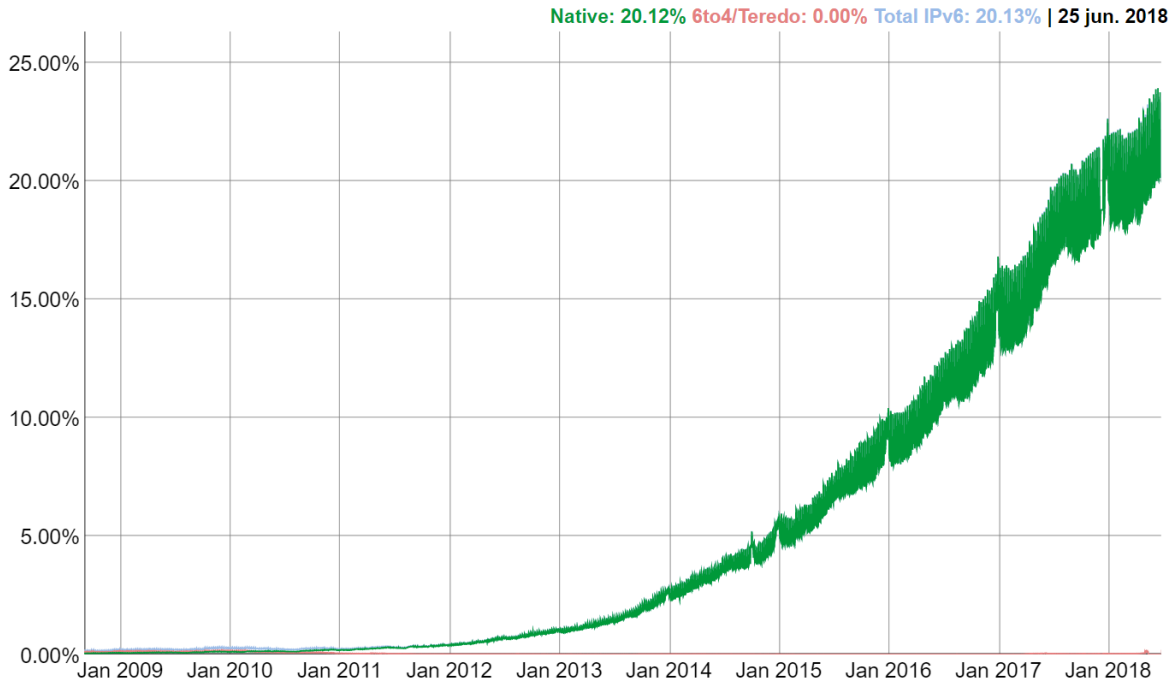


Figura 4 Porcentaje de usuarios que acceden a Google a través de IPv6.

2.6. RED DE SOPORTE DEL IOT

La mayoría de las redes de Internet de las Cosas utilizan IP a partir de alguno de sus puntos. Podemos usar IP desde los propios dispositivos que actúan como sensores o desde la puerta de enlace que concentra la información de varios sensores y luego la envía por una red IP. En cualquier caso se hace uso de este protocolo tan extendido. El actual IPv4 lleva empleándose desde los inicios del Internet moderno y desde algunos años se encuentra saturada por su limitado número de direcciones posibles [6]. IPv6 es la respuesta a este problema que nos permitirá desplegar en un futuro redes masivas de sensores conectados en red, que puedan comunicarse entre sí y con el resto de los servicios en la nube.

IPv6 ofrece una escalabilidad masiva en cuanto al modelo de direccionamiento ya que existen 2^{128} direcciones posibles comparadas con las 2^{32} direcciones que ofrece IPv4. Este número de direcciones hace posible que se puedan abarcar cualquier necesidad de comunicación actual y en el futuro. Otra de las ventajas más importantes de IPv6 es la eliminación del NAT como un elemento necesario. Esto permitirá comunicaciones seguras de punto a punto de manera real. IPv6 también nos permite beneficiarnos de un modelo de enrutamiento más rápido y distribuido. Con IPv6 los nodos pueden autoconfigurase sin la necesidad de un servidor de DHCP al que solicitar la configuración de la red. Por último IPv6 es el sucesor del protocolo de Internet actual lo que significa que permitirá conectar todos los dispositivos y ordenadores de la misma manera con la cual están conectados a Internet en este momento, pero con los nuevos beneficios que aporta IPv6.

El IoT puede beneficiarse del IPv6 al igual que el IPv6 se beneficia del IoT. A continuación daremos algunos argumentos que justifican la afirmación anterior. En primer lugar, la adopción de este nuevo protocolo es inminente y solo es cuestión de tiempo hasta que se globalice su uso en Internet. En la Figura 4, obtenida de la referencia [7], podemos apreciar una aproximación de la adopción mundial de IPv6 a fecha del 25 de junio de 2018. En segundo lugar, nos permite asignar a un mismo dispositivo varias direcciones de red. Podemos asignar una dirección IP para el ámbito local de la red y otra para comunicarnos globalmente con otros nodos. Incluso podemos asignar varias de estas direcciones para ocultar nuestra MAC al resto de la red. En el caso del IoT se han desarrollado versiones reducidas de IPv6, como por ejemplo 6LowPAN, que permite ser utilizado en dispositivos con recursos limitados [8]. Por último con IPv6 podemos asignar una dirección única a un dispositivo y emplearla en cualquier red a la que se conecte. Gracias a esta característica podemos mantener la comunicación con un dispositivo sin importar desde donde esté accediendo a Internet.

Este último punto también supone un riesgo para la privacidad y la seguridad de los usuarios, ya que siempre podremos identificar el mismo dispositivo, aunque este se apague y se vuelva a conectar o aparezca conectado desde otro lugar. Para solucionar este problema podemos contar con varias direcciones IPv6 e ir asignándolas de manera aleatoria o emplear NAT como último recurso [9]. Otro problema que implica la masificación de dispositivos de IoT es la necesidad de actualización de los elementos de red para que puedan soportar todos los nuevos dispositivos que se conectarán a Internet y por tanto la cantidad de tráfico que generarán en un futuro no muy lejano.

En resumen hemos visto como IPv6 permite que el Internet de las Cosas pueda convertirse en una realidad, que nosotros podamos interactuar con los dispositivos que nos rodean de manera sencilla y mejore nuestra calidad de vida.

2.7. REDES INALÁMBRICAS DE SENSORES

En esta sección hablaremos de las redes se sensores actuales, en qué consisten, cuáles son sus componentes principales y cómo se consigue la interoperabilidad entre redes.

Gracias a los avances que trae consigo la miniaturización de la tecnología hoy en día es posible crear nodos equipados con distintos sensores, procesadores e interfaces de radio inalámbricas. Estos pequeños nodos utilizan muy poca energía durante su funcionamiento y están disponibles en el mercado a un precio asequible. La tarea principal de estos nodos es obtener información sobre las condiciones de una zona a través de los sensores y transmitirla de nodo en nodo hasta llegar a un sistema externo.

Un solo nodo sensor tiene capacidades limitadas. Sin embargo, si sumamos el trabajo coordinado de miles o cientos de miles nodos sensores podremos conseguir resultados a gran escala que satisfagan un propósito común dentro de una aplicación de IoT. La distribución de la inteligencia entre toda la red promueve la escalabilidad de esta, ya que el tráfico de control y datos se reduce y distribuye entre toda la red. La distribución de la carga entre varios nodos contribuye a la reducción del consumo energético y así alargar la vida útil de las baterías [10].

Las grandes redes de sensores se caracterizan por contar con una gran densidad de nodos desplegados dentro de una zona determinada. Esa proximidad entre los nodos nos permite que cada nodo pueda comunicarse con varios vecinos. Cuantos más nodos cercanos más caminos se podrán crear y utilizar para la transmisión de información. Por tanto, las transmisiones realizadas podrán utilizar cualquier camino (óptimo o no) para llevar la información desde el nodo donde se genera hasta el nodo que la recoge, en el sumidero de la red. Este modelo de trabajo se diferencia con los esquemas tradicionales, donde el nodo que produce la información transmite directamente la información al nodo sumidero y no tiene en cuenta al resto de los nodos. Además este último modelo emplea mucha más energía para la transmisión y es menos fiable. Una red densa de nodos nos permite reducir esos requisitos energéticos, mientras que conservamos la posibilidad de enviar información a puntos muy lejanos, siempre que la red sea lo suficientemente grande y densa.

Generalmente los nodos se despliegan cerca de la zona donde se produce el fenómeno que queremos capturar. No es necesario que la posición de estos nodos este predefinida, es decir, son capaces de descubrirse entre ellos y formar la red automáticamente. Por ejemplo, las redes de sensores pueden desempeñar alguna de estas tareas: geolocalización de bienes, detección de condiciones ambientales en zonas poco accesibles o monitorización de grandes extensiones de terreno. Las aplicaciones de recolección de datos generan grandes conjuntos de información. Por ello se comprime o se agregan varios datos antes de ser enviados. La comprensión de información se realiza en el nodo empleando operaciones lógicas simples y otros procedimientos que aumenten la eficiencia de las transmisiones. Dependiendo de la aplicación y la naturaleza de los datos podremos emplear unas técnicas u otras para conseguir el resultado óptimo. Todas estas optimizaciones persiguen un objetivo doble: reducir el consumo energético y entregar el mayor número de datos en un solo envío. Actualmente se está trabajando en mejorar este aspecto de las redes inalámbricas de sensores. El software que controla la red ofrecerá al usuario un modo de funcionamiento que alargue la vida útil de las baterías a cambio de reducir las velocidades de transmisión y elevar los tiempos de retardo de los envíos por la red [11].

Aunque han sido presentados diferentes protocolos y algoritmos para las redes Ad hoc tradicionales, estos no reúnen los requisitos necesarios para cumplir con las funcionalidades y los requisitos de las aplicaciones que proponen las redes inalámbricas de sensores. Para entender este problema mostraremos a continuación las diferencias entre las redes inalámbricas de sensores y las redes Ad hoc:

- El número de nodos en una red de sensores puede ser significativamente mayor al número de nodos que compone una red Ad hoc.
- Los despliegues de redes de sensores son muy densos.
- Los nodos de las redes de sensores son más susceptibles al fallo.
- La topología de las redes de sensores puede cambiar frecuentemente.
- Las redes de sensores tienen como objetivo principal difundir la información recolectada, en cambio los dispositivos Ad hoc se utilizan para comunicaciones punto a punto.
- Los nodos de las redes de sensores están limitados en cuestión de consumo energético, capacidades de computación y memoria.
- Los nodos de las redes de sensores no tienen un identificador único global debido a la sobrecarga que supone en redes con un número elevado de nodos.
- Las redes de sensores se despliegan con la intención de obtener información del entorno mientras que las redes Ad hoc se despliegan por propósitos de comunicación entre dos o más dispositivos sin un nodo central.

Teniendo en cuenta las diferencias que hemos señalado en las redes inalámbricas de sensores, las redes de sensores han suscitado el interés de la comunidad científica para proponer nuevas soluciones y protocolos que puedan responder a futuros desafíos.

Por último, reflexionaremos sobre el escalado de las redes inalámbricas de sensores. En primer lugar, podemos dividir un conjunto de sensores en grupos dependiendo del área en la que estén desplegados. En cada subárea habrá un nodo sumidero que se encargue de recoger la información de resto de nodos. Ese nodo sumidero estará en contacto con otros nodos sumideros de otras áreas y todos éstos tendrán conectividad con el sumidero final de la red. En cada subárea podemos emplear cualquier protocolo o tecnología inalámbrica que necesitemos. Esto nos posibilita la creación de redes heterogéneas que abarquen un gran número de dispositivos diferentes pero que persiguen un mismo objetivo. Las subáreas podrán crearse de manera dinámica, añadir o eliminar nodos bajo demanda e incluso fusionarse entre sí según los cambios en la topología de la red. Como los protocolos pueden variar de dispositivo a dispositivo, necesitamos un tipo de nodo híbrido que sea capaz de interactuar con cualquier tipo de dispositivo existente en la red y poder llegar a un acuerdo para intercambiar información.

En resumen, las redes inalámbricas de sensores nos proporcionan información valiosa y una visión más detallada sobre qué es lo que sucede en el entorno que se está estudiando en un instante determinado. En el futuro las redes inalámbricas de sensores evolucionarán hasta convertirse en una parte fundamental en nuestras vidas, incluso más importante que los ordenadores personales actuales [11].

2.8. PLATAFORMAS DE IOT ADICIONALES

A continuación hablaremos de dos plataformas de IoT que se investigaron y probaron antes de comenzar para comprobar su viabilidad para este prototipo. Aunque al final nos decantamos por The Things Network, estas dos plataformas ofrecen funciones muy interesantes que merecen ser mencionadas en este documento. Estas plataformas nos servirán como ampliación en futuros proyectos y como centro de proceso para aplicaciones de IoT.

2.8.1. PYCOM PYBYTES

Pybytes es la plataforma de IoT de Pycom [12] y es parte de su estrategia comercial para ser un proveedor de IoT integro. Pycom es capaz de proporcionarnos desde los dispositivos hardware hasta la plataforma necesaria para utilizar esos dispositivos. Podemos clasificar Pybytes como una plataforma de nivel 1 según la cantidad de características que ofrecen hasta el momento. La funcionalidad básica para administrar y gestionar los dispositivos de ese mismo fabricante se ofrece de manera gratuita como una capa intermedia de una aplicación de IoT. Además ofrece la posibilidad de integrarse con otras plataformas de IoT más avanzadas a través del pago de una cuota mensual por dispositivo. Esto nos permite utilizar las funciones de una plataforma de IoT más completa a la vez que podemos supervisar y gestionar cuidadosamente el estado de cada uno de los dispositivos que tengamos desplegados de este mismo fabricante.

Actualmente el estado de esta plataforma se encuentra en desarrollo y algunas de sus funciones no están terminadas completamente. Por ejemplo dentro del apartado del estado de salud de los dispositivos existen varios campos desactivados y un aviso que indica que la plataforma continúa en desarrollo. Esto significa que los desarrolladores continúan con la creación de la plataforma y que un futuro se añadirán más funcionalidades a la misma.

Para poder usar esta plataforma solo es necesario registrarse, al igual que en cualquier otro servicio en línea, y descargarse el firmware específico para usar la plataforma. Este firmware se instala en las placas y se encarga de conectarse a la plataforma automáticamente usando un código que permite identificar al dispositivo y asociarlo a tu cuenta. La conexión se realiza mediante una red WiFi, las credenciales de acceso a esta red también se solicitan durante la grabación del firmware. Una vez que se ha conectado, el dispositivo envía periódicamente su estado a la plataforma. Desde la consola de la plataforma podemos ver el estado de todos los dispositivos que tengamos asociados e interactuar directamente con ellos. De cada dispositivo podremos ver qué tipo de placa es, si tenemos conectada un módulo de expansión, la versión del firmware instalada y su última conexión a Pybytes. Desde la plataforma podemos enviar ordenes directamente al programa que está ejecutándose en el dispositivo e interactuar con la información que recibimos de él. Existe un apartado en la página del dispositivo que nos permite crear alertas. Por ejemplo podemos definir que aparezca un aviso en la plataforma cuando el nivel de batería se encuentre por debajo de un mínimo o cuando estemos enviando demasiados datos a la plataforma en un corto periodo de tiempo.

Como conclusión podemos decir que esta plataforma de IoT tiene un camino largo por delante y muchos retos que enfrentar para conseguir su hueco en el mercado de las plataformas de IoT, el cual continúa creciendo cada día. Sin embargo el hecho de poseer una integración total con los dispositivos del mismo fabricante le concede una gran ventaja sobre el resto de la competencia ya que Pybytes puede venderse junto con sus dispositivos y ser la opción por defecto cuando utilicemos una plataforma junto a alguna de una de sus placas.

2.8.2. MYDEVICES CAYENNE

La segunda plataforma de IoT que se probó fue Cayenne, un proyecto de la empresa myDevices [13]. Cayenne se define a sí misma como la plataforma de IoT que puedes construir tú mismo. Ofrece una vista de control totalmente personalizable según el dispositivo, el tipo de sensor o actuador que estemos utilizando. Todo esto se configura a través de pequeños módulos de información llamados *widgets*. Además de la aplicación web la plataforma también nos ofrece su aplicación para teléfonos móviles que cuenta con el mismo conjunto de funcionalidades. Esto permite que la plataforma sea más accesible y así poder utilizarla en más entornos. Si empleamos el sistema de clasificación que hemos descrito en este capítulo podemos afirmar que Cayenne cumple con las características de una plataforma de IoT de nivel 2.

Al igual que Pybytes Cayenne se encuentra en desarrollo, pero en un estado mucho más maduro. Actualmente no podemos utilizar todos los *widgets* que están en la documentación y sus aplicaciones comerciales todavía pueden ser muy limitadas. Sin embargo ya tienen soporte para muchas placas orientadas al IoT y que están disponibles en el mercado. Cayenne dispone de un programa para fabricantes que permite a estos integrar su hardware con la plataforma. Esto nos permite trabajar rápidamente con la plataforma, ya que solo será necesario instalar un paquete o subir un programa preparado para ese dispositivo [14]. Si por el contrario deseamos crear nuestra propia aplicación que trabaje con Cayenne podemos hacerlo empleando su API para dispositivos basada en MQTT [15]. Esto nos otorga un grado más de libertad a la hora de desarrollar aplicaciones específicas para el IoT y adaptarlas completamente a los requisitos que demande un cliente o proyecto concreto.

Cayenne nos permite la creación de reglas para ejecutar determinadas tareas cuando uno de nuestros sensores detecte una situación en concreto o programar la ejecución de una determinada acción en un actuador para automatizar el mantenimiento del entorno que estemos controlando. La información que está salvaguardada en la plataforma es exportable a los formatos de bases de datos más populares. En la consola podemos ver en tiempo real la información que está entrado en la plataforma o establecer filtros para recuperar datos de un intervalo de tiempo determinado o conseguir una vista más compresible de los datos. Todas esas tareas que podemos lanzar desde la aplicación web también son accesibles mediante su API. Con la API de Cayenne podemos acceder y modificar dispositivos, reglas y obtener información que esté contenida en la plataforma de manera estructurada. Esto nos permite integrar Cayenne con el resto de software o servicios que utilicemos junto con nuestros dispositivos o con otras plataformas de IoT para construir aplicaciones más avanzadas.

Una característica que también comparte con Pybytes es la capacidad de localizar globalmente los dispositivos que tengamos desplegados y conectados a nuestra plataforma. Gracias a esta funcionalidad podemos consultar en tiempo real la localización de todos nuestros dispositivos y representarlo en un mapa al cual se accede mediante la aplicación web.

La empresa myDevices también es propietaria de la marca IoT in a Box [16] que suministra una solución completa de Internet de las Cosas para servir un propósito determinado dentro del mercado empresarial. Estos kits incluyen una Gateway LoRaWAN, varios nodos con sensores, una solución software para administrar todos esos dispositivos y una consola que una versión adaptada de Cayenne. La gran ventaja de este sistema es que puedes llevar esa consola en el móvil y estar informado de todo lo que ocurre en tu negocio.

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA PROPUESTO

Durante el desarrollo de esta sección hablaremos del diseño que queremos conseguir siguiendo la idea de sumidero híbrido. Este nodo permitirá la creación de una red de sensores con la capacidad de soportar múltiples tecnologías inalámbricas, protocolos y ser escalable cuando las necesidades de la aplicación lo requieran. En primer lugar explicaremos en qué consiste la idea de sumidero híbrido. Posteriormente introduciremos el prototipo derivado de la idea principal que se pretende conseguir en este trabajo como punto de partida para próximos trabajos. Dentro de la implantación hablaremos de los requisitos hardware necesarios y las plataformas que darán soporte a ese diseño. La elección de unos componentes u otros determinarán las prestaciones que podremos esperar de este prototipo. Más tarde hablaremos de las herramientas software que hemos instalado y configurado para que el sistema desempeñe su función. Al final de esta sección se expondrán un conjunto de diagramas que proporcionarán una visión más general y fácil de entender de todo el sistema al completo.

3.1. CONCEPCIÓN DEL DISEÑO

Las redes inalámbricas de sensores actuales están compuestas por dispositivos muy variados que utilizan diversas tecnologías inalámbricas simultáneamente e incluso disponen de varios protocolos de paso de mensajes y de control en el mismo sistema. Las aplicaciones de IoT requieren la cooperación de muchos de estos dispositivos y por ello es necesario que puedan comunicarse entre sí para desempeñar un trabajo en común. Como apunta el título de este trabajo la idea principal que se persigue es el diseño de un sumidero híbrido para redes inalámbricas de sensores, que sea capaz de interactuar con diferentes protocolos y tecnologías inalámbricas. Además será capaz de elegir la tecnología inalámbrica que se utilice en cada momento, según unas métricas que establezcamos. Este sumidero híbrido podrá enrutar los datos que reciba hacia una la plataforma externa, aunque uno de los caminos deje de funcionar. Las interfaces que en un primer momento se utilizan para recibir datos de los sensores pueden convertirse en los caminos de subida para el resto de las interfaces de ese nodo. Esta flexibilidad es la que se pretende conseguir con el desarrollo de este trabajo.

Por ejemplo, un nodo puede utilizar Bluetooth o LoRa para comunicarse con su sumidero. Sin embargo otro nodo puede que solo utilice Sigfox para entregar los datos recolectados. Las diferencias entre los nodos también pueden indicar diferencias entre los protocolos que utilicen. Algunos dispositivos pueden que solo necesiten la pila de protocolos IP, otros pueden que empleen protocolos de paso de mensajes como MQTT o protocolos propietarios. El sumidero híbrido es aquel nodo que podrá interactuar con todos los dispositivos anteriores y configurarse así mismo para enviar cualquier tipo de datos al exterior. Además como contamos con múltiples tecnologías inalámbricas, el nodo sumidero podrá elegir o cambiar la tecnología de comunicación con el otro nodo siguiendo una política de ahorro de batería o la disponibilidad de conexión de estas. También podemos aprovechar esta característica para proporcionar tolerancia a fallos en el caso que la conexión con una tecnología falle y cambiemos automáticamente a otra que esté disponible.

En la Figura 5 se muestra un diagrama que cuenta con tres nodos híbridos que trabajan conjuntamente para transportar los datos desde los sensores hasta el punto de acceso WiFi que tiene conectividad con Internet. El flujo de datos (indicado con flechas azules) va desde los sensores hasta el exterior pasando por los nodos híbridos. En nuestra infraestructura también existe una Gateway LoRa, que se emplea para nodos que solo trabajan con LoRa. En un momento dado nuestro punto de acceso WiFi se desconecta por algún motivo desconocido y la red necesita reconfigurarse para seguir enviando los datos hacia el exterior. Ante esta situación el nodo híbrido intentará buscar conectividad usando otras tecnologías inalámbricas. En este caso tenemos disponible la Gateway LoRa. El nodo aprovechará este nuevo camino y comenzará a enviar la información por él. El resto de la red se ajustará para configurar el nuevo nodo sumidero y encaminar el tráfico hacia a él. La Figura 6 muestra otro diagrama representando esta nueva situación en la que se utiliza LoRa como puerta de enlace con el exterior. En este caso el flujo de información se representa con el color naranja. El cambio de la puerta de enlace es transparente a los nodos sensores. La asociación con los nodos intermedios continúa activa y siguen teniendo conectividad con el exterior.

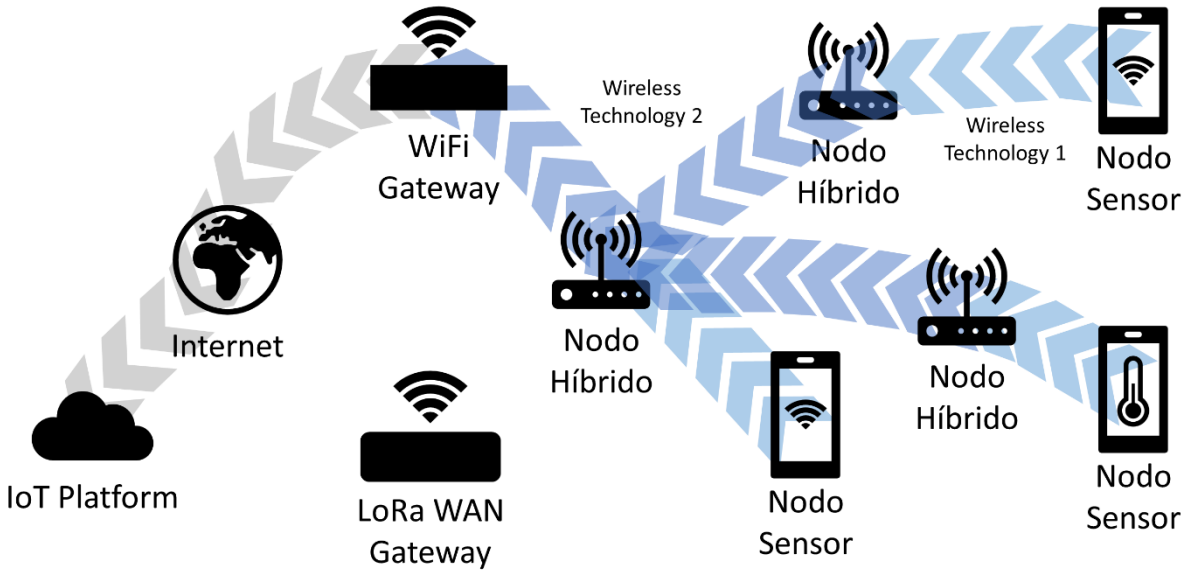


Figura 5 Representación del funcionamiento de la red con nodos híbridos.

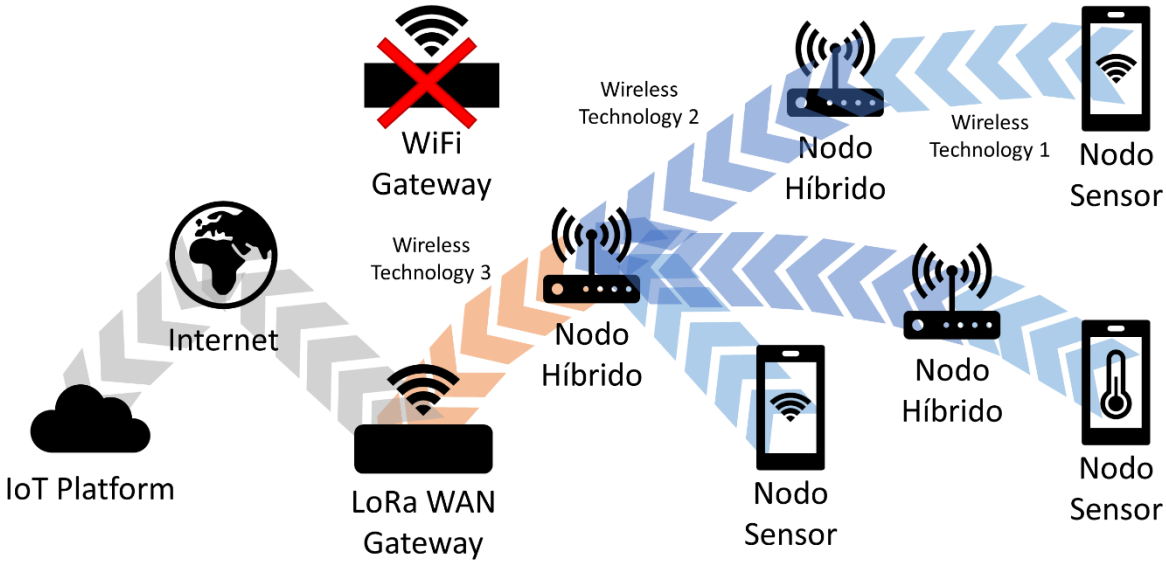


Figura 6 Funcionamiento de la red tras la reconfiguración de los nodos.

La realización de este trabajo supone el punto de partida para la creación de ese sumidero híbrido que se completará en trabajos futuros. Debido a la complejidad del proyecto y el número de pruebas necesario para presentar un sistema funcional que cumpla con esos objetivos, hemos decidido crear una versión reducida de este sumidero. Sin embargo el diseño planteado nace de la misma idea que persigue el proyecto, pero a menor escala.

En primer lugar, el sumidero híbrido contará con un abanico reducido de tecnologías compatibles con el fin de verificar el correcto funcionamiento de estas antes de incluir un mayor número. Para la comunicación interna se identificará cuál es el protocolo ideal sobre el que trabajar como base. A partir de este protocolo se implementarán las interfaces de programación necesarias para la comunicación con el resto de los protocolos que se puedan presentar en el diseño final. Utilizaremos la pila de protocolos IP como medio para el envío de información. Durante el desarrollo del este trabajo hemos configurado una serie de nodos. Estos nodos cuentan con una sola tecnología inalámbrica, en este caso WiFi. Emplearemos esa conexión para crear una red en malla que conecte todos los nodos que existan en la red. El diseño actual nos permite dar conectividad directa a Internet al resto de nodos sensores que se conecten a la red principal. También se ha creado una Gateway LoRaWAN para probar esta tecnología dentro del prototipo actual. El uso de esta Gateway permitirá realizar otros tipos de pruebas que se desarrollarán en el siguiente capítulo. Con el diseño en funcionamiento se realizarán una serie de pruebas para poder verificar el prototipo actual.

En versiones posteriores podremos ampliar las capacidades del sumidero para aceptar más tecnologías inalámbricas como: Bluetooth LE (Low Energy), ZigBee, Sigfox, LTE o NFC (Near Field Communication). En cuanto a los protocolos podemos implementar los siguientes: MQTT (Message Queue Telemetry Transport), HTTPS, ZeroMQ, AMQP (Advanced Message Queuing Protocol), entre otros. Teniendo en cuenta todo lo anterior creemos que esta filosofía de nodo híbrido podría ser una nueva solución innovadora en redes de sensores, que nos permitiría construir redes más complejas y más funcionales que nunca.

3.2. DESCRIPCIÓN GENERAL DEL PROTOTIPO

Como ya adelantábamos el prototipo actual es una versión reducida que sigue el modelo propuesto en las secciones anteriores para la creación de un sumidero híbrido. Contaremos con varios dispositivos que utilizarán una tarjeta inalámbrica para crear una red WiFi en malla. En un futuro el diseño tendrá la capacidad de formar el mismo tipo de red en malla empleando otras tecnologías como Bluetooth LE. A cada nodo se podrán conectar cualquier dispositivo o nodo sensor que sea compatible con WiFi. Esto le proporcionará conectividad con Internet y su tráfico se encaminará por todos los nodos hasta llegar a la Gateway. La puerta de enlace a Internet puede encontrarse en cualquier nodo de la red, el algoritmo de encaminamiento se encargará de que el resto de nodos puedan llegar a este nodo sumidero. La red en malla que hemos configurado se basa en el modo de funcionamiento Ad hoc de la tecnología WiFi. Este modo de funcionamiento no requiere que exista un nodo central que se encargue de asociar y desasociar las estaciones remotas. Cualquier nodo que se encuentre en rango podrá conectarse al resto de dispositivos y formar parte de la red. En futuros diseños intentaremos mantener la filosofía de redes Ad hoc como pilar esencial para otras tecnologías.

Una vez que se tenemos interconectados todos los dispositivos entre sí y con Internet necesitamos un servicio que recoja y nos muestre los datos que van recolectando los sensores. Para realización de este trabajo utilizaremos una plataforma que nos proporcione unas funciones básicas y que a ser posible de manera gratuita. Por el momento nos centraremos en diseñar la red de nodos híbridos, antes de entrar en la configuración de plataformas de IoT más complejas o diseñar aplicaciones que puedan sacar partido a nuestra red. En un futuro el sumidero híbrido contará con diversos protocolos que le permitan comunicarse con las principales plataformas de IoT.

Todos los datos que se generen desde los sensores directamente o a través de un sumidero se transportará por la red en malla hasta una puerta de enlace con Internet. Esta red en malla ha sido configurda empleando HSMM-PI que explicaremos en detalle en el punto 3.5. HSMM-PI dispone de dos modos de funcionamiento, los dispositivos que tengan conexión directa con Internet trabajarán en modo WAN. Esta conexión se realizará por cable, empleando la interfaz Ethernet. La interfaz inalámbrica quedará reservada para la conexión de nodos a la red Ad hoc. El resto de los dispositivos emplearán la interfaz de red inalámbrica para la comunicación con la red en malla, al igual que los equipos anteriores. La interfaz cableada se utilizará para formar una red LAN, es decir, otra red (que podremos convertirla en inalámbrica si es necesario) a la cual se conectarán los nodos sensores. Como se puede observar la mayoría del tráfico pasará por la red WiFi hasta que llegue a una puerta de enlace, donde la conexión se realizará mediante la tecnología Ethernet. A partir de este punto la información viajará por Internet hasta la plataforma de IoT que hemos elegido y configurado.

Dentro de la red en malla el encaminamiento estará gestionado por el protocolo OLSR que se encargará de crear las reglas apropiadas en cada dispositivo e indicar hacia qué nodo se debe de encaminar el tráfico generado para alcanzar Internet. El protocolo OLSR nos permite disponer de más de una puerta de enlace a Internet. Nosotros destinaremos dos dispositivos para que funcionen en modo WAN y nos proporcionen tolerancia a fallos en caso de uno de ellos se desconecte prematuramente o perdamos la conexión con él. Ante cualquier error la red podrá reconfigurarse para seguir permitiendo la conexión con el exterior, simpre que tengamos otra puerta de enlace activa y funcional. En siguiente capítulo realizaremos un conjunto de pruebas para verificar el funcionamiento del sistema.

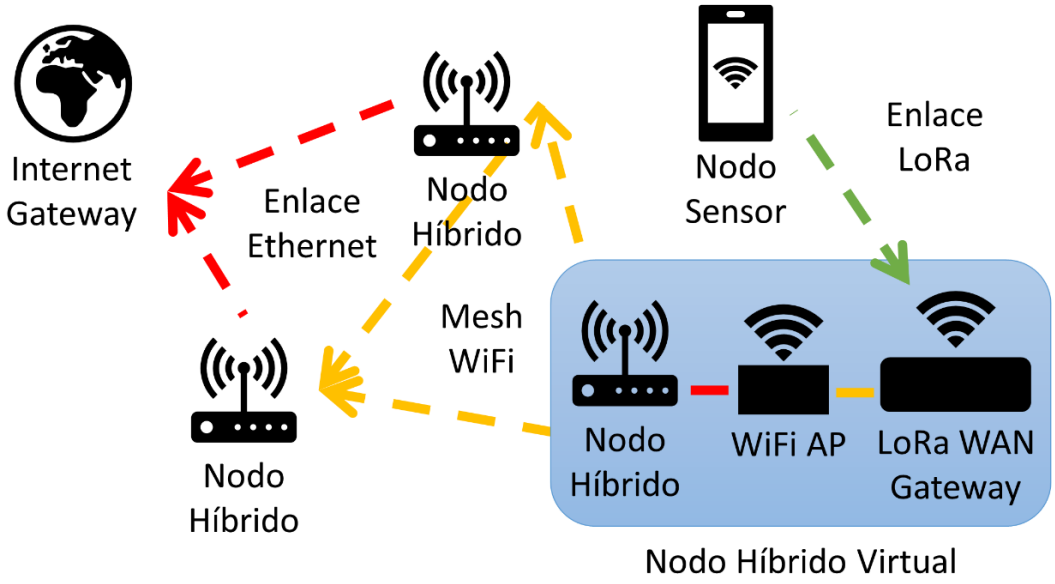


Figura 7 Propuesta de infraestructura para el prototipo.

La Figura 7 muestra un diagrama de la infraestructura que hemos diseñado en este trabajo para implementar el prototipo inicial de sumidero híbrido. En la parte central se encuentran representados nuestros nodos híbridos Estos nodos híbridos solo trabajan con WiFi y tienen conectividad mediante Ethernet. En nuestro caso los nodos no se consideran sumideros ya que es necesario esta conexión cableada para llevar la información fuera de la red. El sumidero híbrido podrá conectarse directamente a Internet empleando cualquier tecnología inalámbrica. A la izquierda se encuentra la puerta de enlace a Internet, a la que se conectan los dos nodos híbridos. En la parte derecha se encuentra otro nodo híbrido que, además de la conectividad WiFi, dispone de una Gateway LoRaWAN. Esta puerta de enlace da servicio a los nodos sensores que utilizan la tecnología LoRa. Para implementar esta última parte del prototipo hemos construido un nodo híbrido virtual, que se compone por varios elementos. En primer lugar encontramos un nodo híbrido que tiene conectado un punto de acceso WiFi, la conexión se realiza mediante Ethernet por cable. Esto ha sido necesario ya que la otra interfaz inalámbrica del nodo está ocupada por la red en malla. En futuros diseños será posible emplear otra interfaz inalámbrica para este propósito y eliminar la necesidad de un punto de acceso externo. Por último a la red inalámbrica que crea el punto de acceso hemos conectado la Gateway LoRaWAN. Esta puerta de enlace LoRa podrá cambiar su funcionamiento y convertirse en módem para dar acceso a Internet al resto de la red si no dispone de otro camino hacia el exterior. En el punto 3.4.4 de este mismo capítulo se explica en qué consiste la tecnología LoRaWAN y cuáles son principales características.

3.3. DISPOSITIVOS HARDWARE EMPLEADOS

Para el montaje de este prototipo necesitamos algunos dispositivos hardware que cumplan con las necesidades de conectividad que hemos descrito en el punto anterior. Estos sistemas deben de poder crear una red en malla Ad hoc, tener un bajo consumo, contar con prestaciones suficientes para ejecutar el algoritmo de encaminamiento OLSR y sobre todo serán sistemas de bajo coste. La documentación en línea disponible también será un factor para tener en cuenta. A continuación explicaremos las características de los dispositivos hardware empleados para la implementación del diseño propuesto.

3.3.1. RASPBERRY 3B

En primer lugar hablaremos del miniordenador Raspberry Pi en su tercera generación modelo B, la cual podemos ver en la Figura 8, obtenida de oficial del fabricante. Fue lanzada al mercado por la Fundación Raspberry Pi, situada en Reino Unido, en febrero de 2016. El objetivo de esta compañía es crear ordenadores de reducidas dimensiones y bajo coste, pero con altas prestaciones enfocadas principalmente al desarrollo de pequeños programas. Gracias a este tipo de proyectos todo el mundo puede aprender a programar de manera divertida y servir como base para la realización de proyectos de mayor envergadura [17].

Concretamente la Raspberry empleada en este proyecto cuenta con las siguientes características hardware:

- **Arquitectura**: ARMv8-A (64/32-bit).
- **Chip**: Broadcom BCM2837.
- **CPU**: 900 MHz 64-bit quad-core ARM Cortex-A53.
- **GPU**: Broadcom VideoCore IV.
- **RAM**: 1 GB (compartido con la GPU).
- **Almacenamiento**: puerto para MicroSDHC (permite el arranque por USB).
- **Conectividad**:
  - I/O: 4 x USB 2.0, 1 x Ethernet 10/100 (via USB).
  - Interfaz de red inalámbrica 802.11n y Bluetooth 4.1.
  - HDMI Rev. 1.3, Vídeo Compuesto a través de la salida TSR.
  - 17 x GPIO, Micro USB para la alimentación de 5 v.
  - Puertos en placa para Webcam CSI y Pantallas DSI sin controlador.
- **Precio de salida**: $35 (al igual que se predecesoras).

Figura 8 Miniordenador Raspberry Pi 3B.

3.3.2. BEAGLEBONE BLACK + WI-PI

La BeagleBone Black es un miniordenador de bajo coste que podemos ver en la Figura 9 obtenida de la web fabricante. Esta placa está disponible en el mercado desde abril de 2013 y ha sido diseñada por la empresa electrónica Texas Instruments. La placa tiene soporte de la comunidad y permite que los aficionados a los proyectos de programación tengan una plataforma donde probar su código fácilmente. Podemos decir que es un competidor de la Raspberry Pi que comparte sus mismos objetivos [18].

Existen varias variantes de este modelo, pero concretamente el nuestro dispone de las siguientes especificaciones:

- **Chip**: AM3358/9.
- **CPU**: Cortex-A8 + Dual PRU (200MHz).
- **GPU**: PowerVR SGX530 (200 MHz).
- **RAM**: 512 MB DDR3.
- **Almacenamiento**: puerto MicroSDHC + 8-bit eMMC de 4 GB
- **Conectividad**:
  - I/O: 1 x USB 2.0, 1 x Ethernet 10/100 (via USB).
  - Toma de entrada de alimentación + un puerto Mini-USB de 5v
  - Puerto Micro HDMI.
  - 4 x UART, 8 x PWM, LCD, GPMC, MMC1, 2 x SPI, 2 x I2C, A/D Converter, 2 x CAN bus y 4 Timers.
- **Precio de salida**: $70.

Para el diseño usaremos dos BeagleBone Black y dos Raspberry Pi B para crear la red en malla y proporcionar conectividad con Internet al resto de la red. Por su parte la Raspberry cuenta con las dos interfaces necesarias una de red inalámbrica y otra alámbrica, sin embargo la BeagleBone solo dispone de esta última interfaz. Por ello hemos decidido conectarle el adaptador de red inalámbrico que podemos apreciar en la Figura 9 [19]. Gracias a este adaptador podemos dotar a esta placa de dos interfaces de red y poder utilizarla dentro del diseño junto con las Raspberry.

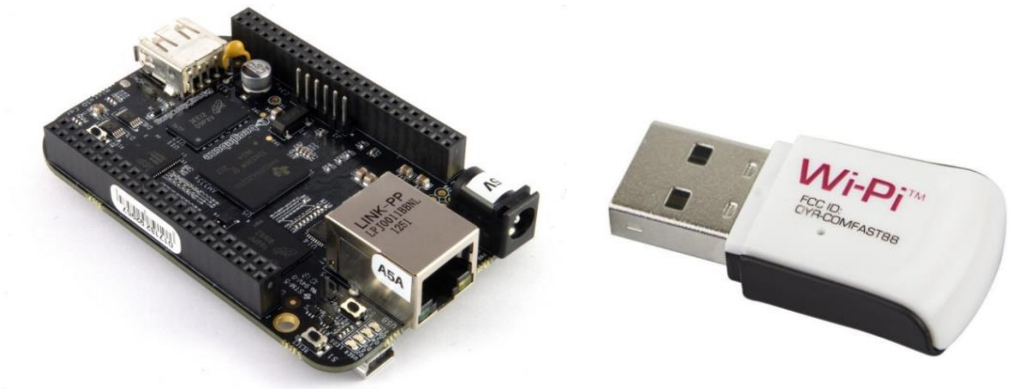


Figura 9 Placa BeagleBone Black y Adaptador Inalámbrico Wi-Pi.

3.3.3. PYCOM LOPY + PYSENSE

Por último hablaremos del módulo LoPy del fabricante Pycom, la cual podemos observar en la Figura 10 extraída de la web del fabricante [20]. Esta placa está diseñada con una visión claramente orientada al Internet de las Cosas y que sus prestaciones lo avalan. Cuenta con un microcontrolador que ejecuta código en MicroPython [21] y es capaz de trabajar con tres tecnologías inalámbricas: Wifi, Bluetooth y LoRa.

A continuación numeraremos las especificaciones técnicas más relevantes:

- **CPU**: Espressif ESP32 chipset.
- **RAM**: 512 KB.
- **Flash**: 4 MB + microSD card.
- 2 x UART, 2 x SPI, I2C, I2S, 4 x Timers, 24 x GPIO.
- Procesador dedicado para el radio Wifi y soporte para la pila IPv6.
- Bajo consumo en funcionamiento y modo de descanso profundo.
- Bluetooth Low Energy y Módulo LoRa (soporta la pila LoRaWAN y es compatible con las frecuencias de funcionamiento de múltiples regiones.

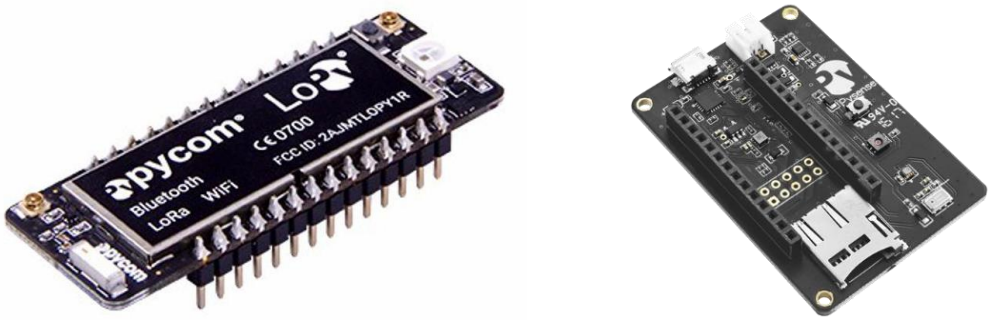


Figura 10 Módulo de IoT LoPy y la placa de expansión Pysense ambos de Pycom.

Para utilizar estos módulos es necesario montarlos sobre una placa de pruebas o adquirir una de las placas de expansión que también fabrica la misma empresa, en este caso hemos optado por la placa Pysense [22], que podemos ver en la Figura 10. Esta placa nos ofrece la posibilidad de montar una tarjeta microSD, alimentar la placa mediante USB o con una batería y además monta una serie de sensores: temperatura, humedad, iluminación ambiental, presión, barómetro y acelerómetro.

En nuestra infraestructura contamos con dos módulos LoPy con sus placas de expansión que actuarán como sensores finales y podremos probar la conectividad de LoRa de una placa a otra y con la Plataforma de IoT. Pycom oferta más placas con más posibilidades de conexión. Por ejemplo tenemos módulos equipados con conectividad LTE [5], Sigfox [3] o NB-IoT [4]. Recientemente se ha lanzado el módulo LoPy4 [23] que añade conectividad Sigfox a la LoPy actual. Además también han desarrollado otras placas de expansión que cuentan con módulos GPS [24] y la posibilidad de instalar dos LoPy u otras placas que desarrollen con el mismo formato [25].

3.3.4. PUNTO DE ACESO LINKSYS EA4500

El Linksys EA4500 es un Router inalámbrico de doble banda con cuatro puertos ethernet y conexión por USB. En la Figura 11, que hemos obtenido de la web del fabricante [26], se encuentra reflejado su aspecto exterior y la conectividad que ofrece.

Gracias a la comunidad de software libre este Router es compatible con una de las distribuciones para Routers más populares. Esta distribución se denomina OpenWRT [27] y permite añadir al equipo muchas más funciones que el fabricante había desarrollado para él en un principio. Todas las versiones de OpenWRT permiten la configuración de un servidor VPN, ajustar opciones ocultas de los radios inalámbricos o elegir como están conectadas internamente las interfaces de red. Todas opciones se pueden configurar mediante línea de comandos o mediante la interfaz web que trae incluida. Otra de las razones por la que utilizar este tipo de software es la seguridad. Los fabricantes pueden ser descuidados y dejar puertas traseras abiertas que sobrepasen la poca seguridad que traen estos dispositivos, permitiendo la entrada de ataques remotos. Este firmware personalizado puede ser ampliado aún más instalando paquetes de software desarrollados por la comunidad. En resumen las posibilidades de este firmware son prácticamente ilimitadas.

Figura 11 Punto de Acceso Linksys EA5400.

Ya sea con el firmware de serie o el personalizado que tiene instalado, este equipo nos ofrece varios modos de funcionamiento (Router, puente, …) y configuraciones avanzadas como el empleo de NAT o la creación de un servidor de DHCP. Sin embargo para la realización de este trabajo nos bastará con activar el modo de punto de acceso Wifi. Esto provocará que se desactive el firewall y cualquier otra funcionalidad y que el dispositivo solo actúe como puente entre el radio WiFi y el puerto Ethernet. Durante las pruebas usaremos este equipo como punto de acceso conectado a una de las placas anteriores por cable. Esto nos permitirá conseguir una infraestructura totalmente sin cables y así conectar los dispositivos finales a una red Wifi segura. En el siguiente capítulo se darán más detalles sobre las pruebas que ejecutaremos usando este equipo y lo compararemos con otros equipos.

3.4. HERRAMIENTAS SOFTWARE UTILIZADAS

En esta sección mostraremos las herramientas software y sistemas operativos que hemos empleado para darle vida al diseño que hemos desarrollado. Por último también echaremos un vistazo a la plataforma de IoT escogida y a algunas de sus funciones.

3.4.1. HSMM-PI

Como ya hemos comentado en puntos anteriores HSMM-Pi es la herramienta que hemos utilizado para la implementación del prototipo propuesto en este trabajo. HSMM-PI comprende un conjunto de herramientas que permiten configurar fácilmente una Raspberry Pi como un nodo HSMM o High-Speed Multimedia [28]. HSMM-Pi permite configurar una red en malla haciendo uso de una interfaz de red inalámbrica y es compatible con la Raspberry Pi y la placa BeagleBone Black, aunque el proyecto puede ser ampliado para funcionar con otros equipos que también utilicen Ubuntu Linux. En esencia HSMM-Pi es un programa que crea una red Ad hoc, utilizando la tarjeta inalámbrica de estos sistemas, y pone en marcha el protocolo OLSR para que el tráfico generado en las redes LAN lleguén hasta los puertos WAN que se encuentran en en los nodos con salida a Internet.

Para la implementación de este programa hemos hecho uso de este repositorio público de GitHub accesible a través del siguiente enlace: https://github.com/urlgrey/hsmm-pi. El proyecto consiste en una aplicación web escrita en PHP que se usa para configurar y monitorizar cada nodo y un script que instala todo lo necesario en el sistema. El script estaba escrito para funcionar en la versión 12.04 ya obsoleta de Ubuntu Linux. Durante la instalación en este proyecto tuvimos que afrontar numerosos retos para conseguir que todos los paquetes y comandos necesarios se ejecutaran correctamente con una versión más moderna de Ubuntu Linux. En la sección 3.4.3 de este mismo capítulo hablaremos sobre la versión de Ubuntu que hemos utilizado y cómo se ha instalado en cada una de las placas.

Una vez que tenemos la aplicacion HSMM en marcha, solo es necesario entrar en la aplicación web de HSMM y decidir qué nodos actuarán como Gateway y cuales lo harán como nodos pasarela. En la Figura 12, obtenida del mismo repositorio, podemos observar el funcionamiento del sistema empleando un nodo Gateway y dos nodos pasarela.

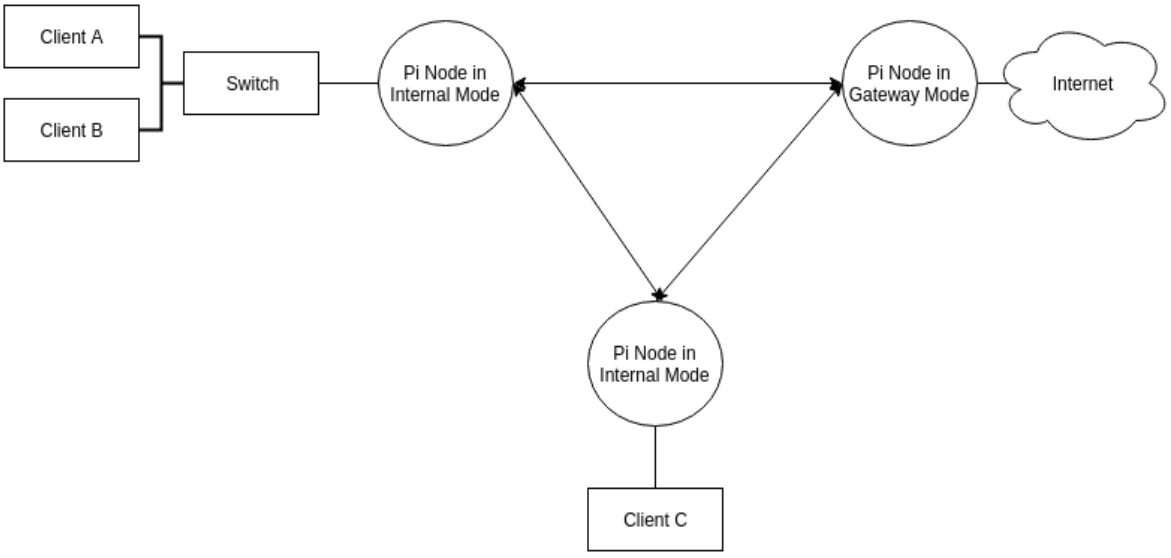


Figura 12 Esquema de funcionamiento de HSMM-Pi.

Cada nodo Gateway utilizará la interfaz Ethernet como puerto de salida a Internet y se configurará como cliente DHCP, es decir, que recibirá su configuración de red de un servidor externo. La interfaz inalámbrica se unirá a la red en malla y podrá contactar con otros nodos que utilicen el mismo nombre de red.

Los nodos pasarela utilizarán la interfaz WiFi para comunicarse con el resto de la red, de igual manera que los nodos Gateway. Sin embargo en este caso la interfaz Ethernet se configurará con una dirección estática y se podrá en marcha un servidor de DHCP y DNS para que los clientes reciban la configuración de red correcta y puedan acceder a Internet.

El encaminamiento entre la red en malla, las redes privadas de cada nodo pasarela y la red de salida de los nodos Gateway se administrará de manera dinámica mediante el protocolo OLSR. Este algoritmo se despliega en los nodos durante la instalación de HSMM-PI. Esto nos evita estar configurando manualmente rutas en cada uno de los nodos y además nos permite una reconfiguración automática en caso de que la topología de la red cambie.

3.4.2. OPTIMIZED LINK STATE ROUTING PROTOCOL (OLSR)

El protocolo OLSR es el sucesor del clásico algoritmo basado en estado del enlace pero optimizado para los nuevos requisitos de las redes Ad hoc móviles. OLSR cuenta con una tabla donde almacena la información de sus vecinos y que intercambia regularmente con otros nodos de la red. Cada nodo de la red elige un subconjunto de sus vecinos como MPRs (multipoint relays) y solo esos nodos son los responsables de reenviar el tráfico de control que se difunde por toda la red. Gracias a este pequeño conjunto de nodos podemos hacer llegar a todos los nodos el estado de la red sin que esta se inunde de mensajes de control. Los nodos MPR también se encargan de construir las rutas más cortas hacia cada destino de la red, como solo lo realizan ellos la cantidad de mensajes es menor y podemos reducir el número de transmisiones totales.

OLSR está preparado para soportar los retos que conllevan las redes móviles y funciona de manera totalmente distribuida, sin la necesidad de un nodo central. En caso de que la red cambie su topología no se requiere volver a calcular las rutas ya que todas las rutas se calculan en un principio y siempre se sabe ir desde de todos los nodos al resto. También puede sobrellevar que la red sea inestable ya que los mensajes de control se retransmiten periódicamente y solo es cuestión de tiempo que los mensajes alcancen todos los nodos. Esto es especialmente útil en redes como la nuestra donde se emplean tecnologías inalámbricas en la que pueden ocurrir colisiones o puedan aparecer interferencias puntuales.

OLSR es independiente del protocolo IP que se utilice y no requiere que se modifique ningún campo de la pila de protocolos IP. OLSR solo interactúa con las tablas de encaminamiento de cada nodo y se encarga de formar las rutas más cortas hasta los nodos Gateway. Además no necesita una transmisión de paquetes ordenada, ya que cada paquete cuenta con un contador que se incrementa para cada mensaje. De esta manera los receptores del mensaje pueden identificar qué información es más reciente aunque los mensajes se hayan intercambiado durante la transmisión [29].

Si los mensajes OLSR se van a difundir por una red abierta podemos proteger ese tráfico mediante el cifrado con una clave pre-compartida. Esto permite que solo nuestros nodos puedan conocer la topología completa de la red y no se filtre al resto.

3.4.3. UBUNTU LINUX

Ubuntu es un Sistema gratuito y de código abierto basado en la distribución Debian Linux. Se encuentra en desarrollo continuo desde 2004 por Canonical y su comunidad de desarrolladores. Canonical proporciona actualizaciones de seguridad garantizadas a cada versión de Ubuntu desde todo su ciclo de vida. El modelo de negocio de Canonical es la venta de servicios de soporte a empresas que utilicen Ubuntu y necesiten soporte inmediato. Al contrario que otros sistemas operativos empresariales, las versiones de la comunidad y la versión empresarial comparten las mismas características [30].

Actualmente Ubuntu se encuentra en la versión 18.04 LTS y distribuye en múltiples versiones: ordenadores personales, servidor, servidores en la nube, dispositivos de IoT y puede ser adaptado a más plataformas. Ubuntu es el sistema operativo más popular en servicios de Cloud Computing y es el sistema de referencia en OpenStack [31].

En nuestro trabajo emplearemos Debian como sistema operativo base para los miniordenadores Raspberry y las BeagleBone. Concretamente emplearemos Raspbian [32] y la versión 14.04 de Ubuntu respectivamente. Raspbian es una distribución adaptada a las características de la Raspberry y optimizada para la arquitectura ARM. En el caso de la BeagleBone hemos empleado una versión también personalizada para este tipo de placas. En el directorio Bibliografía del CD que acompaña a este trabajo encontramos los manuales que hemos seguido para instalar estos sistemas operativos y configurarlos por primera vez.

3.4.4. LORAWAN GATEWAY

La implementación del diseño propuesto incluye la instalación de una Gateway LoRaWAN a la que llegarán los datos de varios sensores que también emplean la tecnología LoRa. El protocolo LoRaWAN permite la comunicación bidireccional de dispositivos de bajo consumo con una Gateway [33]. La Gateway hace de puente entre la radio LoRa y el protocolo IP. Podemos enviar mensajes a través de multicast para hacer llegar mensajes a múltiples dispositivos a la vez o realizar actualizaciones de firmware en masa. El estándar LoRaWAN se ha diseñado para los fabricantes puedan implementarlo libremente y construir sus propias aplicaciones para el Internet de las Cosas [34].

LoRaWAN tiene tres modos de operación según la aplicación del dispositivo:

- **Clase A**: Estos dispositivos están optimizados para un ahorro de energía máximo gracias a las transmisiones asíncronas. El dispositivo decide cuando empezar la transmisión y después escucha durante dos cortos periodos de tiempo para permitir una comunicación bidireccional.

- **Clase B**: Estos dispositivos están sincronizados con la red usando mensajes periódicos y huecos de tiempos prefijados para un retardo en la comunicación que pueda ser calculado. Como inconveniente el dispositivo empleará más energía debido a los mensajes intermedios de sincronización.

- **Clase C**: Esta clase es similar a la clase A, sin embargo el receptor del dispositivo final siempre está activo, lo que permite a la Gateway estar siempre en contacto si recibe nuevos datos para ese dispositivo. Al igual que la clase B mantener la radio encendida supone un consumo energético extra pero a cambio podemos comunicarnos con el dispositivo con la mínima latencia.

Para maximizar la vida de las baterías y la utilización de la red LoRaWAN permite alterar la velocidad de transmisión de datos según nuestras necesidades. Las Gateway podrán emplear varios canales a la vez para evitar que los clientes con distintas velocidades interfieran entre ellos. Las velocidades de LoRaWAN oscilan entre 0.3 Kbps hasta 50 Kbps.

La Figura 13, obtenida desde la web de LoRa Alliance, muestra la arquitectura de funcionamiento de un nodo LoRa, una Gateway que recoge y envía los mensajes y la aplicación que procesa dichos mensajes. Como podemos observar en la misma figura los mensajes se transmiten de forma segura desde el dispositivo final hasta la plataforma. La Gateway solo transporta los mensajes pero no los procesa. La autenticación de los nodos puede realizarse de manera automática (OTAA), es decir, cuando un dispositivo nuevo se conecte con una Gateway recibirá su configuración para establecer un vínculo seguro o también es posible configurar previamente los nodos (ABP) con esa configuración y evitar pasos adicionales durante la conexión [33].

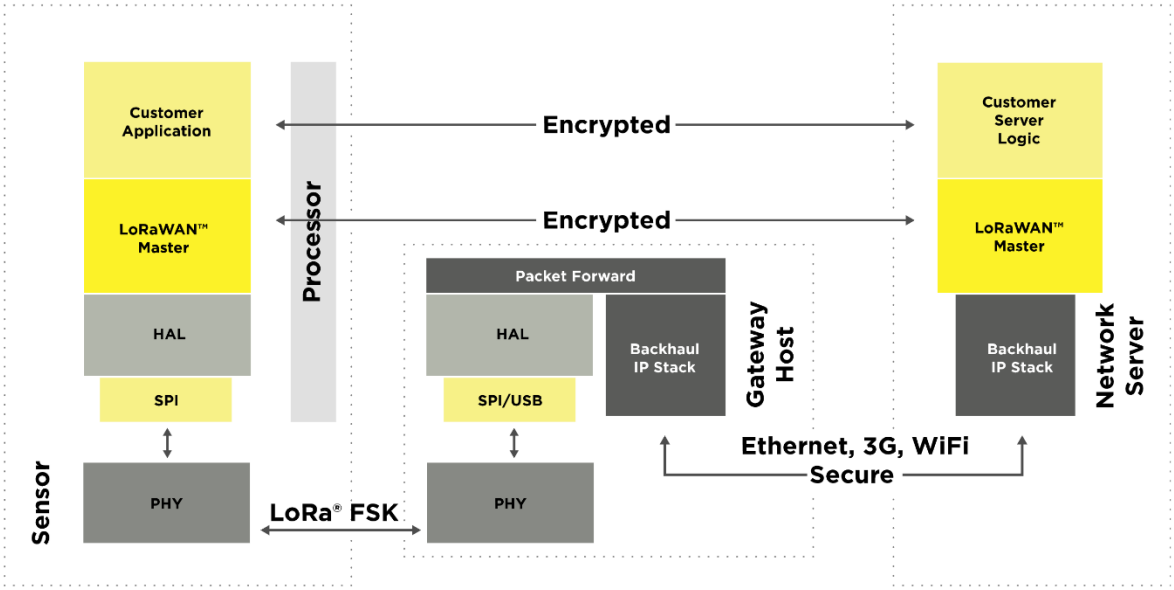


Figura 13 Arquitectura de Funcionamiento de LoRaWAN.

Como comentamos en el punto anterior vamos a utilizar las placas LoPy, que tienen un cometido muy específico y sólo ejecutan programas desarrollados en una versión reducida de Python denominada MicroPython [35]. MicroPython es una implementación ligera y rápida de Python 3 [36] que está optimizada para ejecutarse en un microcontrolador. MicroPython fue un proyecto que fue financiado gracias a la plataforma de Crowdfunding Kickstarter [37] y está disponible mediante la licencia de código abierto MIT. MicroPython se encarga de que los recursos disponibles en el microcontrolador estén disponibles para el programa que está en ejecución.

Para la implementación de la Gateway usaremos el código de ejemplo que está publicado en la web Pycom [38] y en su repositorio de GitHub [39]. De momento no hemos realizado ningún cambio sustancial al código, aparte de modificar los campos necesarios para configurar la Gateway. En trabajos posteriores para conseguir el cambio entre Gateway y módem LoRa será necesario añadir código o grabar un nuevo programa bajo demanda.

3.4.5. THE THINGS NETWORK

The Things Network [40] es una plataforma para el Internet de las Cosas orientada principalmente a los dispositivos que utilicen LoRaWAN para conectarse con Internet. Siguiendo la clasificación de plataformas de IoT que se hizo en el capítulo 2, podemos incluir esta plataforma en el grupo tecnológico de nivel 1. Esto supone que necesitamos la ayuda de otra plataforma de IoT para almacenar y procesar los datos que recibe desde los sensores. Sin embargo como uno de los objetivos de este tarbajo es verificar la viabilidad de este prototipo, TTN nos proporciona las funciones suficientes como para comprobar que los datos que enviamos desde los sensores son recibidos correctamente en sus servidores.

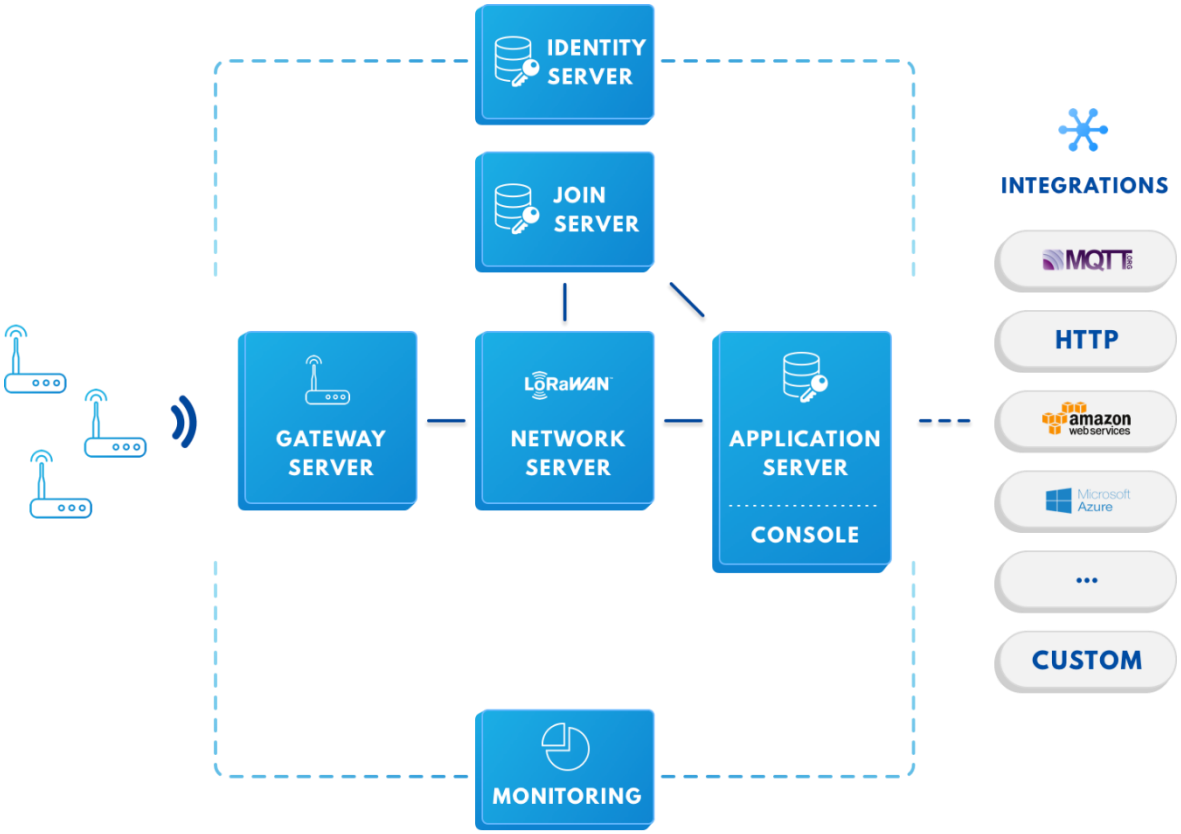


Figura 14 Pila de servicios que ofrece The Things Network.

La Figura 14, extraída de la referencia [41], muestra un diagrama con todos los componentes de la plataforma. Empezando por la izquierda encontramos el primer componente fundamental, los Gateway Servers. A estos servidores llegan todos los datos que reciben desde las Gateways de los usuarios. Las Gateway permiten la comunicación entre la plataforma y los dispositivos finales. Podemos emplear diversos dispositivos para crear nuestra propia Gateway y ponerla en línea públicamente. Desde la plataforma podemos ubicar globalmente esa Gateway para que usuarios cercanos puedan utilizarla para sus aplicaciones. Una vez que los mensajes han llegado a la plataforma, éstos pasan por un servidor (Network Server) que desempaqueta su contenido y lo transmite al servidor de aplicaciones (Aplication Server). En la parte de arriba se encuentran los servidores de inicio (Join Server) que se encargan de controlar qué dispositivos están autorizados a formar parte del sistema como Gateways y qué dispositivos pueden conectarse a ellas. Las claves de inicio de sesión de todos los nodos se encuentran almacenadas en los servidores de identidad (Identity Server).

Además el servidor de identidad administra los perfiles de usuario de la plataforma y asegura que sólo los usuarios autorizados puedan acceder a los recursos de la plataforma y a la gestión de nodos y Gateway. En la parte derecha podemos observar que el servidor de aplicaciones está conectado con otras plataformas de IoT para exportar e importar la información que los sensores envían a la plataforma, o enviar mensajes de vuelta a ellos. Esto nos permite integrar TTN en una infraestructura de IoT, para formar aplicaciones complejas que necesiten de varias fuentes de datos. Algunas de esas plataformas de IoT son las compañías de servicios en la nube más populares como Amazon Web Services [42] o Microsoft Azure [43], pero también podemos interactuar con los recursos de TTN empleando una API u otro protocolo de paso de mensajes con MQTT [44]. Por último el módulo de monitorización y diagnóstico se utiliza para proporcionar al usuario una vista de control de todo lo que está sucediendo con sus dispositivos y herramientas para diagnosticar cualquier problema. Todos estos módulos están implementados a modo de microservicio que pueden ser replicados y escalados según las necesidades de la plataforma. Si en un momento dado el sistema experimenta un pico de tráfico anormal, se pueden poner en marcha más instancias del servicio saturado y así poder responder eficazmente a las demandas que se produzcan [41].

TTN utiliza UDP como protocolo de transporte de datos y el puerto 1700 para las comunicaciones con la Gateway. Los mensajes llegan al servidor por medio de un socket UDP que crea la Gateway LoRaWAN y los mensajes se envían en texto plano.

Analizando todas las características anteriores hemos decidido utilizar esta plataforma por tres razones principalmente. En primer lugar TTN tiene una facilidad de uso inigualable para agregar una nueva Gateway o un nuevo dispositivo. Para configurar uno de estos sistemas nos bastará con introducir un nombre y la plataforma nos proporcionará todos los datos necesarios para que el dispositivo se conecte a ella. En la Figura 15 se puede observar la consola de TTN con nuestra Gateway LoRaWAN registrada y configurada correctamente. En segundo lugar TTN es una plataforma que principalmente trabaja con dispositivos LoRaWAN, lo que la hace ideal para probar este prototipo. Por último TTN nos permite conectar nuestra aplicación con otras plataformas de IoT y con ello podremos integrar el resto de tecnlogías futuras con otras plataformas una vez que se añadan al prototipo actual.

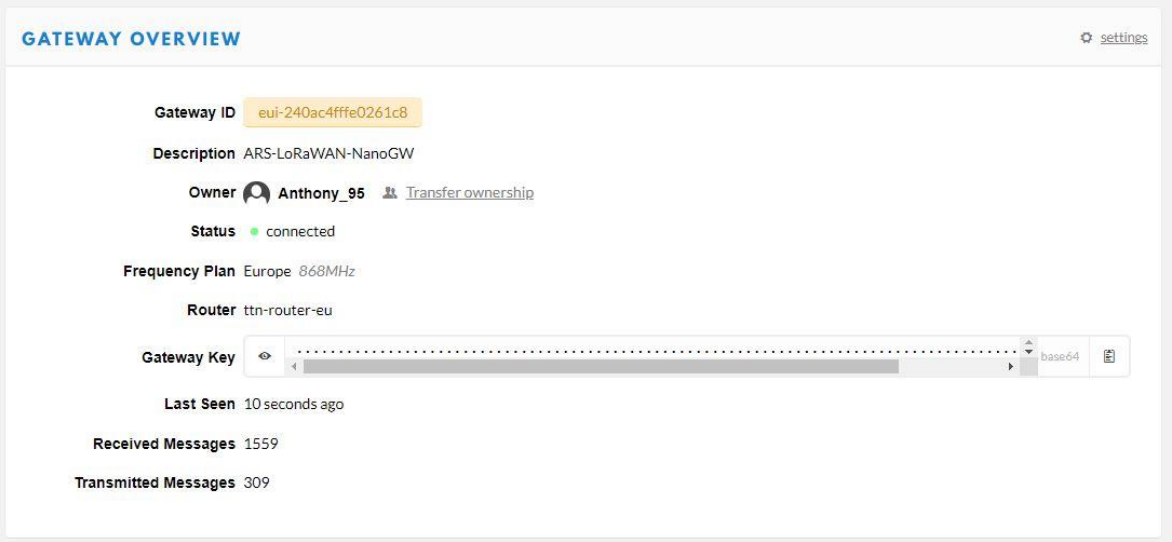


Figura 15 Gateway LoRaWAN configurada en The Things Network.

3.5. FUNCIONAMIENTO DEL PROTOTIPO

Con todos los elementos hardware y las herramientas software explicadas podemos pasar a describir el funcionamiento del prototipo propuesto en conjunto y comprobar como todas esas piezas encajan en el diseño final del sumidero híbrido.

El propósito de este prototipo es desempeñar dos funciones principalmente. En primer lugar el prototipo ha de ser capaz de configurarse automáticamente para levantar la red en malla, mediante WiFi, y permitir que el tráfico generado en las redes LAN pueda alcanzar Internet. En segundo lugar se quiere comprobar si el hardware LoRa es capaz de conectarse a la plataforma TTN e interactuar con ella enviando y recibiendo mensajes. En el punto anterior hemos mostrado el funcionamiento de la Gateway LoRaWAN. En este punto probaremos que los clientes LoRa pueden usar esta Gateway para llegar a la plataforma. Para demostrar esto observaremos si los mensajes de prueba son visibles finalmente a la consola de The Things Network. Una vez que ambos casos estén en funcionamiento combinaremos ambas partes. Como resultado obtendremos una red capaz de comunicarse entre los dispositivos que estén al alcance mediante WiFi, formar una red en malla y proporcionar un canal de subida hacia la red para aquellos sensores que utilicen LoRa.

El funcionamiento de la red será el siguiente: Los dispositivos sensores se conectarán a los puntos de acceso WiFi o a la Gateway LoRaWAN. Dentro de la red los mensajes serán encaminados por los nodos híbridos hasta la puerta de enlace de Internet, que en nuestro prototipo está implementada como una conexión Ethernet. Los datos llegarán finalmente a TTN empleando la conexión por cable a Internet. El sumidero híbrido final será capaz de utilizar ese canal de subida LoRa como canal de subida de toda la red. En caso de pérdida del primer enlace de subida, ya sea Ethernet o WiFi. Toda la red se reconfiguraría para utilizar esa interfaz LoRa como salida a Internet. Además se podrán utilizarán otras tecnologías inalámbricas con el mismo propósito, permitir la entrada de mensajes de los sensores o servir como puerta de enlace de toda la red. La Figura 16 muestra gráficamente todos los componentes que forman parte del prototipo actual. Como se puede observar se encuentran todos los dispositivos se mencionaron anteriormente en la Figura 7. Cabe destacar que los dispositivos que actúan como puerta de enlace de la red en malla están conectados a TTN a través de Internet, no directamente.

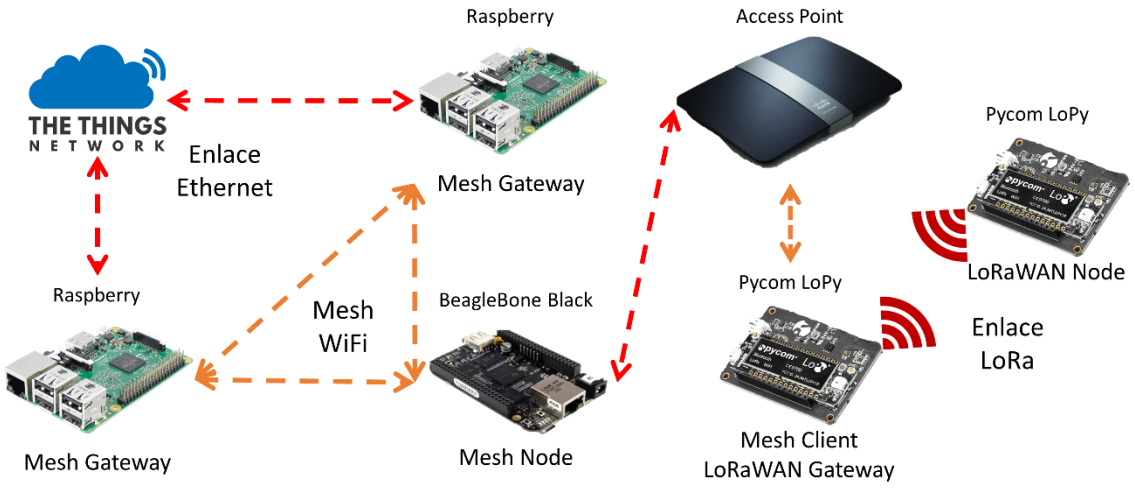


Figura 16 Representación del prototipo actual con el hardware definitivo.

El primer paso para la implementación del prototipo es configurar los nodos híbridos para formar la red en malla. Esto conlleva la instación del sistema operativo y la instalación del programa HSMM-PI. El sistema operativo que se va a instalar es Ubuntu Linux, del cual hablamos en el punto 3.5.3. El programa HSMM-PI es tan sencillo de instalar como ejecutar el archivo instalador que va incluido con él. Después de arreglar algunos de sus comandos para adaptarlo a las versiones de Linux que estamos utilizando el programa se instala correctamente. Una vez que el sistema se encuentra totalmente configurado es hora de ponerlo en marcha y comprobar que efectivamente tenemos conectividad desde uno de los nodos finales hasta la Gateway de la red en malla. A partir de ahí comprobaremos que tenemos acceso a Internet y que podemos contactar con la plataforma The Things Network.

La asignación de direcciones IP en la red en malla se realiza de manera automática y resulta en la creación de una dirección de red única que está basada en los últimos tres fragmentos de la dirección MAC del adaptador inalámbrico de cada nodo. En la Tabla 1 se encuentran recogidas las direcciones IP de la red en malla para cada uno de los nodos junto con su nombre, dirección MAC y el tipo de placa utilizada. Las dos placas Raspberry Pi se han configurado en modo WAN, ya que tendrán conectividad directa con Internet. Al poseer dos sistemas en modo Gateway podemos realizar una serie de pruebas que se desarrollarán en el capítulo siguiente.

Dispositivo	Nombre Asignado	Dirección MAC	Dirección IP
Raspberry Pi	Black	B8:27:EB:69:64:CF	10.105.100.207
Raspberry Pi	White	B8:27:EB:C3:58:FF	10.195.88.255
BeagleBone Black	Blue	00:0F:54:10:23:D5	10.16.35.213

Tabla 1 Direcciones IP de los nodos de la red en malla.

Con respecto al nodo que actúa como pasarela se ha configurado una nueva red local para los dispositivos clientes que tiene la siguiente dirección 172.27.2.0/24. Esta red cuenta con su propio servidor de DHCP y DNS. Cualquier dispositivo que conectemos obtendrá la configuración necesaria para llegar a cualquier nodo de la red. Otra posibilidad es conectar los nodos directamente a la red que forma la malla. Sin embargo esta red no dispone de servicio DHCP y por tanto la configuración de red deberá ser ajustada de forma manual.

El punto de acceso que creará la red de acceso para los dispositivos finales tiene la dirección 172.27.2.10, se ha asignado de manera estática y está fuera del rango de direcciones que concede el servidor DHCP de esa misma interfaz. El rango de direcciones comprende desde la dirección 172.27.2.50 hasta la 172.27.2.75. Recordemos que el punto de acceso sólo actúa como puente y no realiza ningún tipo de gestión en el tráfico de los dispositivos finales. En el Anexo 1 se numeran todos los parámetros de configuración que hemos elegido.

Con el sistema en marcha podemos observar que la configuración es correcta y todos los nodos de la red en malla pueden verse entre ellos. En la Figura 17 se muestra una captura de la interfaz web donde aparecen conectados los nodos que están dentro del rango.

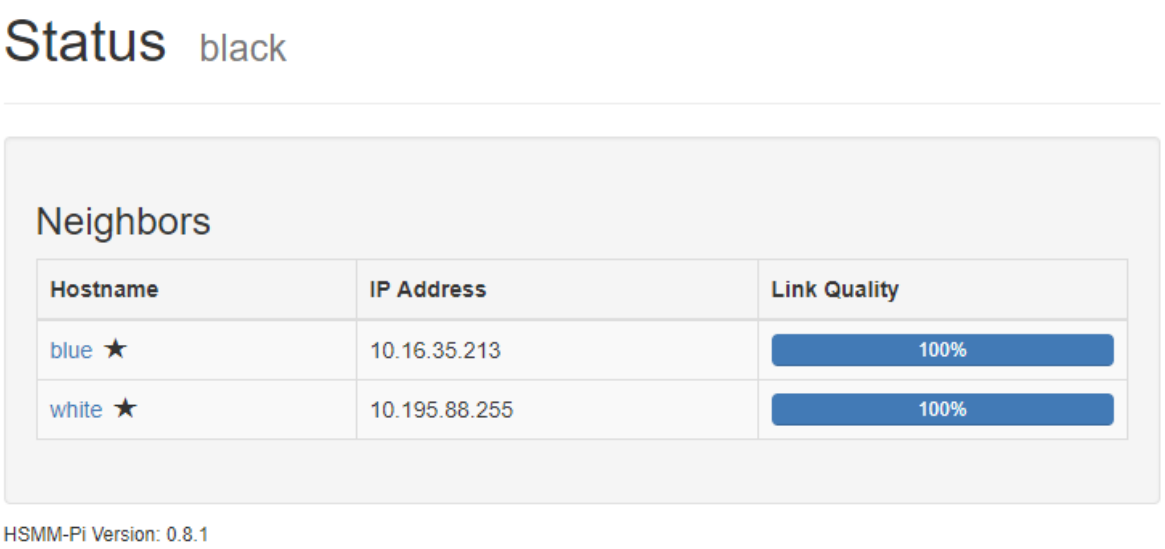


Figura 17 Interfaz Web de HSMM-Pi.

La Figura 18 muestra la tabla de encaminamiento del nodo *Blue*. Este nodo es el que actúa como puente entre la Gateway de la red en malla y el resto de los dispositivos, incluyendo la Gateway LoRaWAN que hemos instalado. En esta figura, además de las redes directamente conectadas a este nodo, se puede observar que el nodo *White* es la puerta de enlace principal para este nodo. En la Figura 19 podemos observar las reglas de firewall que permiten que exista conectividad entre las diferentes redes de la infraestructura. Finalmente podemos probar que el encaminamiento y la red funciona ejecutando el comando PING desde este nodo hasta otro equipo que esté dentro o fuera de la red. Si obtenemos respuesta del nodo remoto significa que efectivamente tenemos conectividad.

```
ubuntu@blue:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.195.88.255   0.0.0.0         UG    2      0        0 wlan0
10.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 wlan0
10.105.100.207  10.105.100.207  255.255.255.255 UGH   2      0        0 wlan0
10.195.88.255   10.195.88.255   255.255.255.255 UGH   2      0        0 wlan0
172.27.2.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
ubuntu@blue:~$
```

Figura 18 Tabla de encaminamiento en la red en malla.

```
root@blue:/home/ubuntu# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@blue:/home/ubuntu#
```

Figura 19 Tabla de reglas IPTables en la red en malla.

Para finalizar este punto comprobaremos que los clientes LoRa pueden alcanzar la plataforma TTN empleando nuestra Gateway LoRaWAN o cualquier otra Gateway pública de TTN que tengamos cobertura. Lo primero será grabar el programa del cliente en la placa LoPy. Este programa está disponible en las mismas librerías de Pycom. El segundo paso es registrar nuestro dispositivo en la plataforma. Este proceso consiste en crear una nueva aplicación y copiar los datos que nos proporciona TTN en el fichero de configuración del cliente LoRa. La Figura 20 muestra la página de configuración dispositivo en la plataforma.

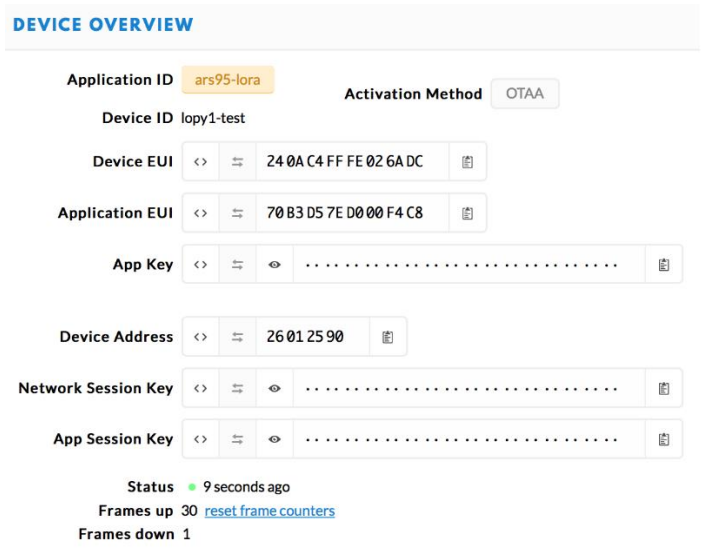


Figura 20 Ventana de configuración del nodo LoRa cliente.

Al encender el nodo por primera vez se empezarán a enviar mensajes de asociación. Las Gateways harán llegar estos mensajes al servidor de TTN. Este decidirá si el cliente está registrado en la plataforma y le enviará la información de inicio de sesión. Esto es necesario ya que se utiliza el método de autentificación OTAA, del cual hablamos en la sección 3.5.4. La Figura 21 muestra el tráfico recibido de nuestra Gateway cuando conectamos el cliente. En la figura anterior también se puede observar el recuento de mensajes enviados y recibidos y la última vez que se recibió un mensaje desde ese cliente. Una vez comprobado que los mensajes del nodo alcazan la plataforma podemos verificar que el cliente LoRa funciona.

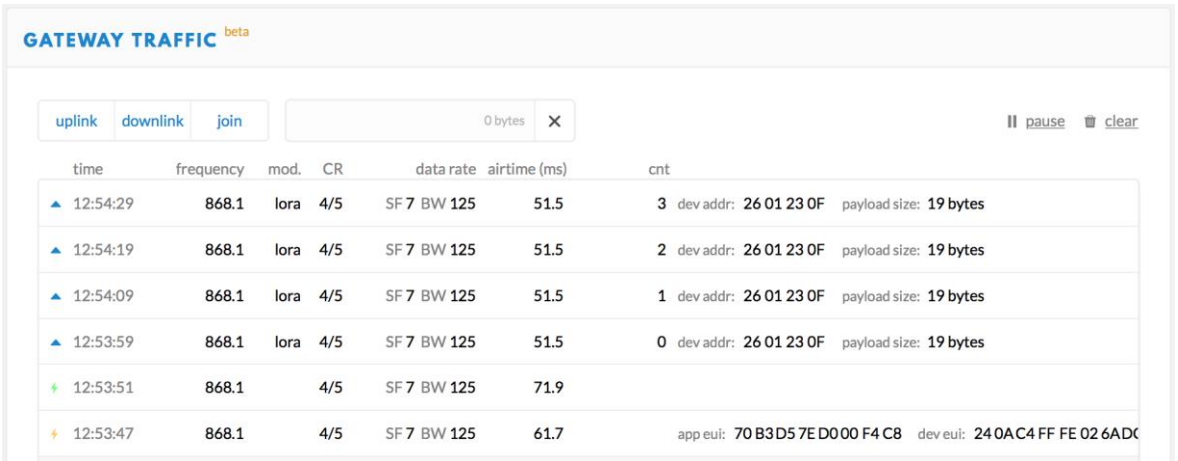


Figura 21 Captura del tráfico recibido por la Gateway LoRaWAN.

CAPÍTULO 4. REALIZACIÓN DE PRUEBAS AL SISTEMA

En este cuarto capítulo hablaremos de todo lo relacionado con la realización de las pruebas al prototipo realizado. Comenzaremos relatando en qué consisten las pruebas que se desarrollarán, detallaremos el entorno dónde se realizarán dichas pruebas y describiremos cómo se ha monitorizado el sistema para comprobar que las pruebas se realizan correctamente. Al final de cada tipo de prueba se mostrarán los resultados recogidos junto a las conclusiones extraídas.

La realización de estas pruebas nos permitirá analizar el funcionamiento de este prototipo en un despliegue real y cómo responde ante diferentes situaciones cuando la integridad de alguno de los dispositivos se ve comprometida. Ya que el sistema implementado es un prototipo de un sistema mucho más complejo, las pruebas certifican que este diseño funciona, pero a una escala mucho menor en comparación con los objetivos de este proyecto. A medida que introduzcamos nuevas tecnologías y protocolos en nuestro prototipo serán necesarias más pruebas. Incluso será necesario realizar pruebas más extensas y detalladas para saber qué sucede en cada momento dentro del sistema.

4.1. ENTORNO DE EJECUCIÓN DE LAS PRUEBAS

Todas las pruebas que hemos definido serán ejecutadas sobre el mismo entorno de operación. Este entorno consiste en un terreno plano de unos 100 m^2. Esto posibilita que podamos probar la cobertura de cada tecnología inalámbrica y no existan interferencias entre ellas. Las tarjetas de red integradas en las Raspberry Pi y el módulo Wi-Pi no alcanzan grandes distancias, aunque tengan vista directa entre ellas. Esto se debe a la tecnología empleada en la antena de radio, que al estar integrada en la placa ofrecen unas prestaciones muy limitadas. Por tanto situaremos cada nodo de la red a una distancia no superior a 10 m del otro como punto de partida para asegurarnos que las dos placas pueden verse entre ellas. El punto de acceso estará junto al nodo pasarela conectado por cable. A 20 m de este situaremos la Gateway LoRa que irá equipada con antenas externas para ambas tecnologías, WiFi y LoRa. El punto de acceso cuenta con antenas internas que amplían notoriamente su cobertura y podremos situar las placas LoPy mucho más lejos que los dispositivos anteriores.

El nodo cliente LoRa podremos situarlo en el punto más alejado del terreno ya que puede abarcar largas distancias sin problemas de conectividad. Esto es una de las ventajas de la tecnología LoRa, ya que los módulos pueden cubrir grandes extensiones de tierra empleando antenas omnidireccionales estándar. Los nodos en malla que actúen como sumidero estarán conectados por cable a un conmutador que proporcionará conectividad con Internet.

Todas las pruebas partirán con la misma configuración de red que hemos explicado anteriormente. Las dos placas Raspberry Pi se configurarán con HSMM-PI para ser nodos sumideros, denominados Black y White. La placa BeagleBone Black, denominada Blue, se configurará como pasarela. Esto creará una red adicional donde se conectarán los clientes y nodos sensores. En nuestro caso esta red es necesaria para proporcionar una configuración de red válida a la placa LoPy. La placa LoPy está configurada para crear una Gateway LoRaWAN y necesita conectividad con Internet. Esta conexión se empleará para reenviar los paquetes que recibe mediante el módulo LoRa a los servidores de The Things Network y emitir por radio las respuestas que le lleguen de vuelta hacia los sensores. La Figura 22 muestra gráficamente cómo estarán situados los nodos en las pruebas. El nodo Blue será siempre el nodo que utilicemos para ejecutar los comandos que nos dan la información sobre qué nodo actúa como Gateway en cada momento y el estado de la red en malla.

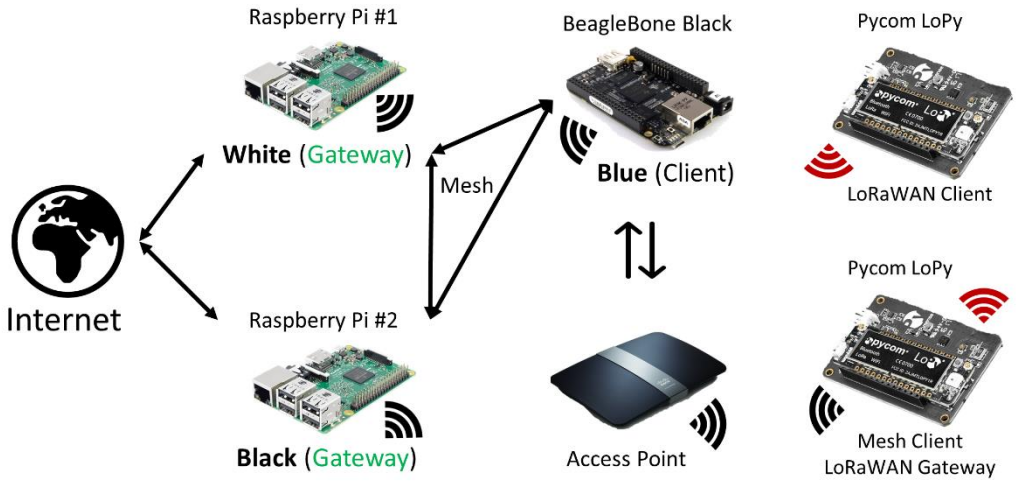


Figura 22 Diagrama de conexionado de los nodos para las pruebas.

4.2. MONITORIZACIÓN DEL SISTEMA DURANTE LAS PRUEBAS

La monitorización del sistema se realizará de manera manual a través de los comandos que ya hemos utilizado para asegurarnos que todo el sistema funciona correctamente. Emplearemos principalmente la interfaz web de HSMM para ver que todos los nodos de la red en malla se han conectado a la red y saber con qué potencia son detectados por el nodo cliente. Para saber en cada momento que host se está utilizando como puerta de enlace emplearemos el comando `route` y `traceroute` para observar el camino que siguen los paquetes por la red. También emplearemos el comando `ping` para medir el tiempo que pasa hasta que un equipo se inicia completamente y comienza a atender peticiones, así como para contar el número de paquetes perdidos durante la prueba de tolerancia a fallos de los nodos sumidero. Para las pruebas de puesta en marcha se utilizarán las marcas de tiempo que muestran los registros del sistema. Dichos registros son accesibles a través del comando `DMESG`.

4.3. TIEMPO DE PUESTA EN MARCHA

4.3.1. DESCRIPCIÓN DE LA PRUEBA

En primer lugar, el objetivo de esta prueba es determinar el tiempo de puesta en marcha de todos los dispositivos del sistema en frío, es decir, desde que encendemos la alimentación hasta que llega el primer mensaje a la plataforma de IoT The Things Network. Esta prueba es especialmente útil para determinar el tiempo mínimo que tiene que transcurrir desde que el sistema se reinicia completamente debido a un fallo o se restablece como consecuencia de un mantenimiento programado. A través de esta prueba podemos conocer qué dispositivos son los más lentos en inicializarse y así centrar nuestros esfuerzos en mejorar sus tiempos de inicio o reemplazarlos por otros más rápidos.

4.3.2. RESULTADOS DE LA PRUEBA

Esta prueba consiste en calcular el tiempo que transcurre desde que de cada dispositivo se pone en marcha hasta que empieza a dar servicio al resto de la red. Una vez que se ha calculado el tiempo individualmente hemos obtenido el gráfico de la Figura 22.

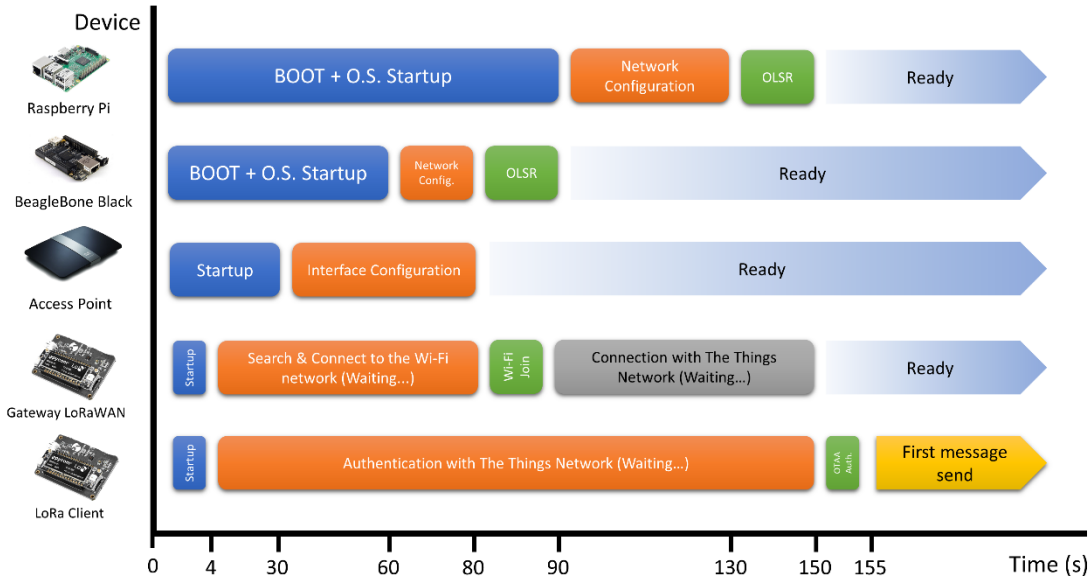


Figura 23 Cronograma de los tiempos de inicio de los dispositivos.

Como podemos observar en esta figura el dispositivo que tiene el arranque más lento es la Raspberry Pi, en total emplea más de dos minutos en conectarse a la red en malla y dar servicio al resto de sistemas. Los tiempos de inicio de la BeagleBone y el punto de acceso son aceptables, teniendo en cuenta el tipo de sistema que son y sus prestaciones. Por último las placas LoPy son los dispositivos más rápidos de este sistema, ya que apenas completan su arranque en 3 segundos y comienzan inmediatamente a ejecutar la aplicación programada. En conclusión podemos decir que los nodos híbridos que utilicemos en un futuro tendrán un tiempo de inicio similar o superior al observado en las placas Raspberry Pi, ya que necesitaremos una plataforma capaz de trabajar con varias tecnologías inalámbricas a la vez. Además el sistema requerirá más potencia de proceso y memoria para desplegar un sistema inteligente que tome las decisiones sobre qué tecnología emplear o qué camino seguir para llevar la información al exterior.

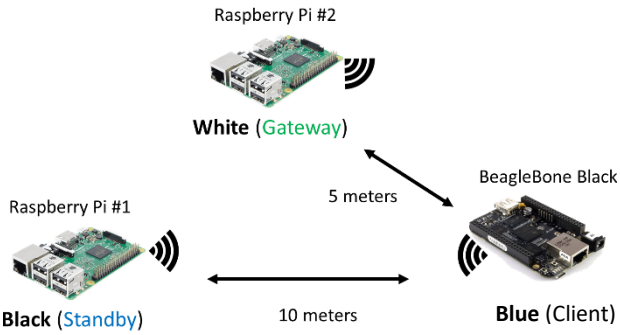
4.4. ELECCIÓN DE LA PUERTA DE ENLACE

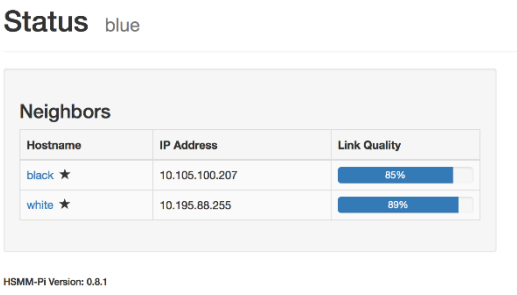
4.4.1. DESCRIPCIÓN DE LA PRUEBA

En segundo lugar, esta prueba evaluará la capacidad del prototipo actual para establecer la puerta de enlace de un nodo según la potencia del enlace que detecte con otro nodo sumidero. Los cambios en la intensidad de la señal provocarán que la selección de la puerta de enlace cambie entre las posibilidades disponibles. Esto significa que si tenemos dos nodos sumidero y uno es detectado con más potencia que el otro, el primero será elegido puerta de enlace principal del nodo cliente. En el caso de solo detectar un nodo sumidero, este siempre sería configurado como puerta de enlace independientemente de la señal detectada. Para la realización de esta prueba contaremos con dos nodos que actúan como sumideros y el nodo de prueba. Posicionaremos los dos primeros nodos de manera que uno de ellos se encuentre más cerca del nodo de prueba que el otro y por tanto tendrá más potencia que el segundo. Comprobaremos qué nodo se usa como puerta de enlace y cambiaremos de posición los nodos sumidero para que el nodo que se encontraba más alejado sea ahora el que se detecte con más potencia. El primer nodo lo situaremos en la posición del segundo nodo. La prueba se completará con éxito si al cambiar de posición los nodos, la puerta de enlace del nodo cliente cambia automáticamente al nodo que esté más cerca en cada momento.

4.4.2. RESULTADOS DE LA PRUEBA

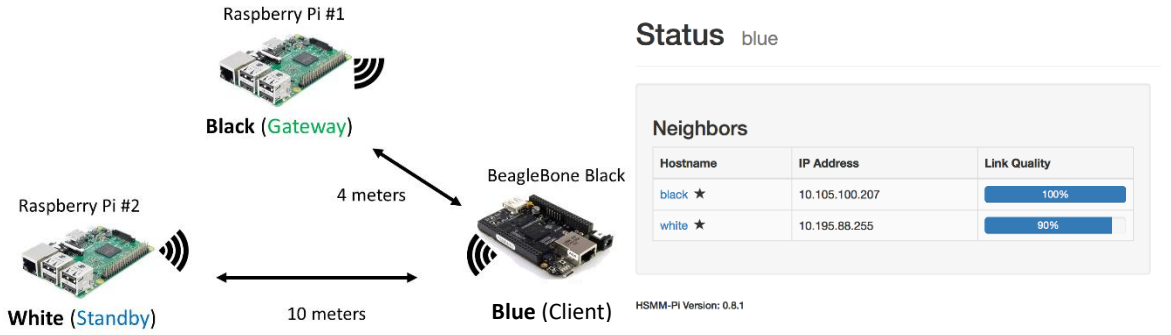
Para esta prueba solo utilizaremos los nodos en malla que muestra la Figura 24. Los dos nodos actúan como posibles Gateways (Black y White) y el otro nodo (Blue) actúa como pasarela y proporciona acceso a Internet a los clientes que tenga conectados a él. Inicialmente el nodo White es seleccionado como puerta de enlace predeterminada ya que es el nodo que más potencia tiene de los dos posibles. El objetivo de la prueba es que el nodo Blue cambie su puerta de enlace al detectar otro nodo sumidero con más potencia que el actual. Si alteramos el escenario de acuerdo con la Figura 25 y dejamos pasar unos minutos podremos observar que las tablas de encaminamiento del nodo Blue ahora señalan al nodo Black como puerta de enlace, ya que ahora tiene más potencia que el nodo White. A la vista de los resultados podemos concluir que la prueba ha sido exitosa.






```
ubuntu@blue:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.195.88.255   0.0.0.0         UG    2      0        0 wlan0
10.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 wlan0
10.105.100.207  10.105.100.207  255.255.255.255 UGH   2      0        0 wlan0
10.195.88.255   10.195.88.255   255.255.255.255 UGH   2      0        0 wlan0
172.27.2.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
```

Figura 24 Situación de partida de la prueba de elección de Gateway.




```
ubuntu@blue:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.105.100.207  0.0.0.0         UG    2      0        0 wlan0
10.0.0.0        0.0.0.0         255.0.0.0       U     0      0        0 wlan0
10.105.100.207  10.105.100.207  255.255.255.255 UGH   2      0        0 wlan0
10.195.88.255   10.195.88.255   255.255.255.255 UGH   2      0        0 wlan0
172.27.2.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
```

Figura 25 Situación final de la prueba de elección de Gateway.

4.5. RECUPERACIÓN ANTE UN ERROR EN UN NODO

4.5.1. DESCRIPCIÓN DE LA PRUEBA

En tercer lugar, determinaremos el tiempo de respuesta del algoritmo de encaminamiento y del sistema cuando un nodo que se está empleando como puerta de enlace desaparece de la red (se desconecta o sufre un error). Cuando esto sucede el resto de los nodos necesitan reconfigurarse para emplear otro nodo sumidero, si existiera otro nodo sumidero y estuviera al alcance. Al igual que el escenario anterior, nuestra red cuenta con dos nodos sumidero y otro nodo que da acceso al resto de clientes. Partiremos con los tres nodos conectados a la red. El nodo cliente habrá seleccionado uno de los dos nodos sumidero como puerta de enlace. Comprobaremos cuál de los dos nodos se está usando como puerta de enlace y le quitaremos la alimentación. La prueba consiste en comprobar que el nodo de prueba es capaz de detectar que ha perdido la conectividad y cambiar al otro nodo sumidero. Contaremos el tiempo que transcurre desde que desconectamos la alimentación y el sistema recupera la conexión con el exterior.

4.5.2. RESULTADOS DE LA PRUEBA

Esta prueba parte del escenario expuesto en la Figura 26, donde, al igual que la prueba anterior, podemos apreciar un nodo cliente (Blue) y dos nodos sumidero (Black y White). Con el sistema en marcha hemos obtenido la puerta de enlace que está empleando el nodo Blue, en este caso el nodo White. Por tanto si desconectamos el nodo Black la tabla de encaminamiento no registraría ningún cambio y la prueba no se podría realizar. A continuación utilizamos el comando `ping` para probar la conectividad con Internet mientras que desconectamos el nodo White (Figura 26). Tras **50 segundos** de pérdida de paquetes los clientes recuperan la conectividad con Internet. En la Figura 27 podemos observar cómo se pierde la conectividad con el exterior cuando desactivamos el nodo White (Gateway). En el cambio solo se pierden 47 paquetes. La conexión con el servidor externo se recupera usando el nodo Black, que se encontraba activo pero no se utilizaba como puerta de enlace.

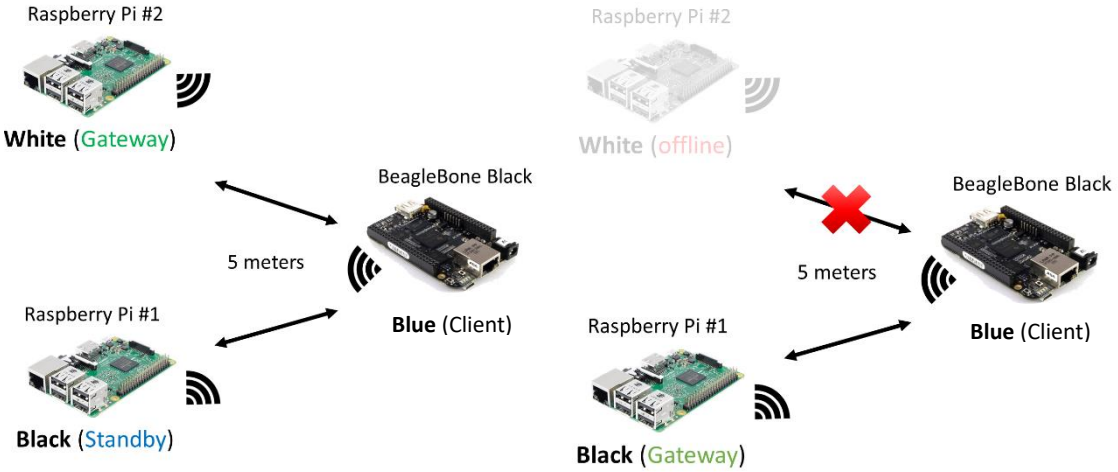


Figura 26 Situación de partida (izq.) y final (der.) de la prueba de reconfiguración.

En mitad de la transición podemos observar que se producen varias respuestas desde el nodo Blue, que indican que no existe una ruta para alcanzar la red solicitada. Estos mensajes se producen como fruto de la reconfiguración de la red, ya que el algoritmo de encaminamiento OLSR debe actualizar las tablas de encaminamiento del host. Durante la reconfiguración OLSR se encarga de buscar una nueva puerta de enlace válida y de propagar los cambios por toda la red. Si ejecutamos ahora el comando `route -n` podemos observar como la puerta de enlace ha cambiado al nodo Black. A la vista de los resultados también podemos indicar que esta prueba es exitosa y podemos certificar que nuestro prototipo puede adaptarse a los cambios de la topología.

```
64 bytes from 216.58.201.131: icmp_seq=206 ttl=54 time=72.650 ms
64 bytes from 216.58.201.131: icmp_seq=207 ttl=54 time=68.673 ms
64 bytes from 216.58.201.131: icmp_seq=208 ttl=54 time=181.208 ms
64 bytes from 216.58.201.131: icmp_seq=209 ttl=54 time=55.776 ms
64 bytes from 216.58.201.131: icmp_seq=210 ttl=54 time=43.987 ms
Request timeout for icmp_seq 212                          Desconexión del Nodo White
Request timeout for icmp_seq 213
Request timeout for icmp_seq 214

--- Acortado ---

Request timeout for icmp_seq 238
Request timeout for icmp_seq 239
Request timeout for icmp_seq 240                          Reconfiguración de la red
92 bytes from 172.27.2.1: Destination Net Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 5400 b2c2   0 0000  40  01 77da 172.27.2.51  216.58.201.131

Request timeout for icmp_seq 241
92 bytes from 172.27.2.1: Destination Net Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 5400 cb54   0 0000  40  01 5f48 172.27.2.51  216.58.201.131

Request timeout for icmp_seq 242
Request timeout for icmp_seq 243
Request timeout for icmp_seq 244

--- Acortado ---

Request timeout for icmp_seq 257
Request timeout for icmp_seq 258
Request timeout for icmp_seq 259                          Red reconfigurada con la nueva Gateway
64 bytes from 216.58.201.131: icmp_seq=260 ttl=54 time=35.959 ms
64 bytes from 216.58.201.131: icmp_seq=261 ttl=54 time=68.007 ms
64 bytes from 216.58.201.131: icmp_seq=262 ttl=54 time=70.866 ms
64 bytes from 216.58.201.131: icmp_seq=263 ttl=54 time=51.014 ms
64 bytes from 216.58.201.131: icmp_seq=264 ttl=54 time=59.484 ms
```

Figura 27 Captura del comando PING de la prueba de reconfiguración.

4.6. ELECIÓN DE UNA NUEVA INTERFAZ COMO GATEWAY

4.6.1. DESCRIPCIÓN DE LA PRUEBA

Esta prueba verifica que alguna interfaz inalámbrica puede reconfigurarse bajo demanda. Disponemos de dos Gateways una WiFi y otra LoRaWAN pública. Actualmente la red se encuentra conectada a la Gateway WiFi, hacia donde se dirige el tráfico, mientras que la Gateway LoRaWAN permanece a la espera. El resto de los nodos también incorporan la tecnología LoRa a través de las placas LoPy, pero están configurada para servir a los sensores que la necesiten. La situación de partida está representada en la Figura 28.

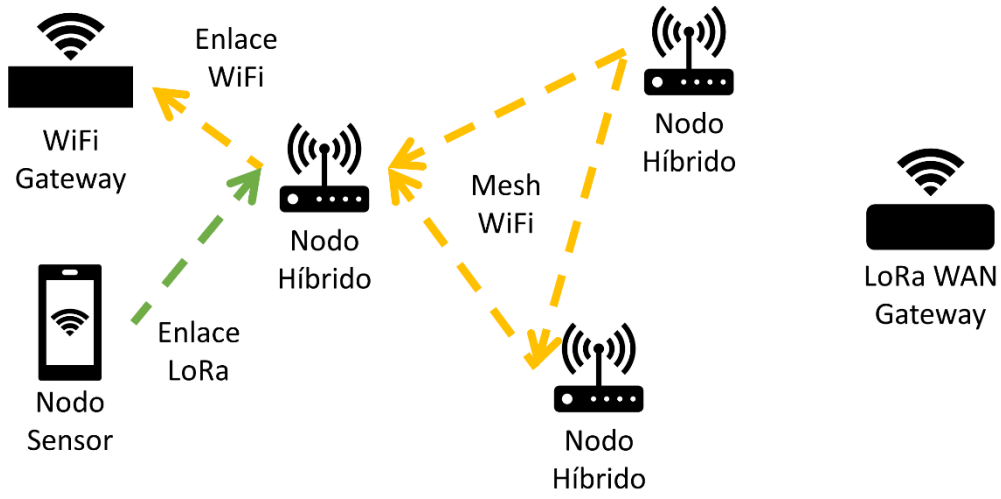


Figura 28 Situación de partida de la prueba de cambio de Gateway.

En un momento dado los nodos deciden que se necesita cambiar de Gateway. En este momento una de las interfaces LoRa que se encargaba de servir a los clientes LoRa se reconfigura para conectarse a la Gateway LoRaWAN pública. Este cambio permite que la red pueda seguir encaminando paquetes al exterior redirigiendo el tráfico por el nuevo camino hacia la Gateway LoRaWAN. La Figura 29 muestra la situación final de la red una vez que se ha completado la migración de Gateway.

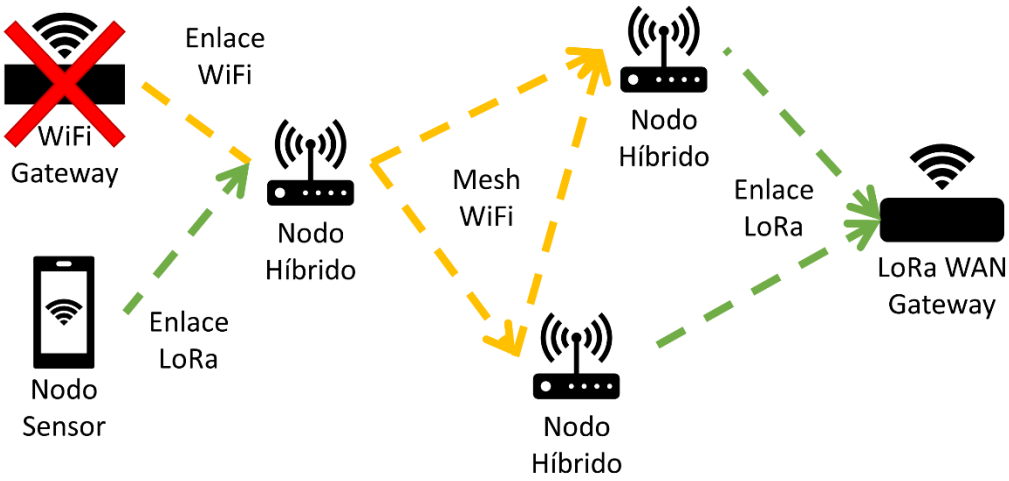


Figura 29 Situación final de la prueba de cambio de Gateway.

CAPÍTULO 5. CONCLUSIONES Y PROPUESTAS FUTURAS

Actualmente las infraestructuras que sirven el tráfico de las redes del Internet de las Cosas necesitan evolucionar para poder adaptarse a los nuevos retos de movilidad, tolerancia a fallos y escalabilidad que las aplicaciones actuales requieren. Este trabajo realiza una serie de aportaciones para solucionar algunos de los retos planteados. Durante la realización del trabajo nos hemos dado cuenta de que una solución flexible es la clave para poder superar esos retos y alcanzar un sistema robusto. La reconfiguración de los nodos de la red es fundamental para responder a los retos que plantean las redes inalámbricas. Con las mejoras que introduce este trabajo se consegue disminuir notablemente los tiempos de inactividad de los sistemas IoT. Los tiempos de mantenimiento se reducen, ya que nuestra infraestructura sigue funcionando a pesar de que un enlace se encuentre inoperativo. Y sobre todo se evita la pérdida de información prolongada durante estos fallos. En definitiva el diseño propuesto ha contribuido en incremetrar la confiablidad de los sistemas de IoT, es capaz de combinar diferentes tecnologías inalámbricas disponibles y establecer rutas alternativas hacia Internet, adaptando su topología a los cambios que surjan durante el funcionamiento.

Como propuestas futuras podemos destacar que nuestro prototipo implementará más tecnologías inalámbricas como Bluetooth, Sigfox, LTE o ZigBee, y nuevos protocolos como MQTT, HTTPS, ZeroMQ o AMQP. Estas integraciones incrementarán las capacidades de reconfiguración y permitiría interactuar con nuevos dispositivos y servicios. Ante un gran abanico de tecnologías tendremos que dotar al sumidero con la capacidad de elegir qué ruta seguir. Las decisiones se tomarán siguiendo distintas políticas, como el consumo energético, el coste por conexión, el ancho de banda que ofrezca cada tecnología o estado actual de los nodos, teniendo en cuenta la batería restante, la cobertura disponible o los requisitos de la aplicación que se esté utilizando.

En resumen, todos los avances que se presentan con relación al Internet de las Cosas, incluyendo este trabajo de fin de máster, contribuyen al mundo del mañana donde todo y todos estaremos conectados al resto de “cosas”. Las aplicaciones de IoT, si se utilizan correctamente, nos ayudarán a hacer que nuestras vidas sean más fáciles.

BIBLIOGRAFÍA

Las referencias en línea que aparecen en esta bibliografía han sido accedidas por última vez el 6 de agosto de 2018.

[1] D. Evans, «The Internet of Things (White Paper),» Cisco Internet Business Solutions Group (IBSG), 2011.

[2] IoT Analytics GmbH, «IOT PLATFORMS,» IoT Analytics, 2015.

[3] Sigfox S.A., «Sigfox Technology Overview,» Sigfox , 2018. [En línea]. Available: https://www.sigfox.com/en/sigfox-iot-technology-overview.

[4] R. R. a. A. G. Nitin Mangalvedhe, NB-IoT Deployment Study for Low Power Wide, Nokia Bell Labs, 2016.

[5] I. Grigorik, High Performance Browser Networking, O'Reilly Media, 2013.

[6] ARIN Ltd., «ARIN IPv4 Free Pool Reaches Zero,» American Registry for Internet Numbers, 2015. [En línea]. Available: https://www.arin.net/vault/announcements/2015/20150924.html.

[7] Google LLC., «Adopción de IPv6,» Google, 2018. [En línea]. Available: https://www.google.com/intl/es/ipv6/statistics.html.

[8] J. Olsson, «6LoWPAN demystified,» Texas Instruments, 2014.

[9] IoT6, «IPv6 advantages for IoT,» FP7 European Research, 2014. [En línea]. Available: https://iot6.eu/ipv6_advantages_for_iot.

[10] W. S. Y. S. E. C. I.F. Akyildiz, Wireless sensor networks: a survey, Elsevier Science B.V., 2001.

[11] K. M. S. T. Z. C.S. Raghavendra, Wireless Sensor Networks, Springer, 2006.

[12] Pycom Limited, «Pybytes the IoT middleware platform for all connected devices,» Pybytes , 2018. [En línea]. Available: https://pycom.io/pybytes/.

[13] myDevices, Inc., «Cayenne The world's first drag-and-drop IoT project builder,» Cayenne, 2018. [En línea]. Available: https://mydevices.com/cayenne/features/.

[14] myDevices, Inc., «IoT Ready Program™,» myDevices, 2018. [En línea]. Available: https://mydevices.com/cayenne/docs/iot-ready-program/.

[15] myDevices, Inc., «Cayenne MQTT API,» myDevices, 2018. [En línea]. Available: https://mydevices.com/cayenne/docs/cayenne-mqtt-api/.

[16] myDevices, Inc., «IoT in a Box – Turnkey IoT Solutions,» myDevices, 2018. [En línea]. Available: https://www.iotinabox.com/product-tour/.

[17] Raspberry Pi Foundation, «Raspberry Pi Model B,» Raspberry Pi Foundation, 2018. [En línea]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[18] BeagleBoard Foundation, «BeagleBone Black,» BeagleBoard , 2018. [En línea]. Available: https://beagleboard.org/black.

[19] Premier Farnell Limited, «WIPI - WLAN USB Module,» Element14, 2018. [En línea]. Available: http://my.element14.com/element14/wipi/wlan-module-for-the-raspberry/dp/2133900.

[20] Pycom Limited, «LoPy LoRa, WiFi and Bluetooth development board,» Pycom, 2018. [En línea]. Available: https://pycom.io/hardware/lopy-specs/.

[21] George Robotics Limited, «MicroPython,» Damien George, 2018. [En línea]. Available: https://micropython.org/.

[22] Pycom Limited, «Pysense,» Pycom, 2018. [En línea]. Available: https://pycom.io/hardware/pysense-specs/.

[23] Pycom Limited, «LoPy4 LoRa, Sigfox, WiFi and Bluetooth development board,» Pycom, 2018. [En línea]. Available: https://pycom.io/hardware/lopy4-specs/.

[24] Pycom Limited, «Pytrack - multi-network module,» Pycom, 2018. [En línea]. Available: https://pycom.io/hardware/pytrack-specs/.

[25] Pycom Limited, «Expansion Board 2.0,» Pycom, 2018. [En línea]. Available: https://pycom.io/hardware/expansion-board-2-0-specs/.

[26] Belkin International, Inc., «Router inalámbrico Smart Wi-Fi de doble banda N900 Linksys EA4500,» Linksys, 2018. [En línea]. Available: https://www.linksys.com/es/p/P-EA4500/.

[27] OpenWrt/LEDE Project, «OpenWrt,» 2018. [En línea]. Available: https://openwrt.org/about.

[28] ARIN Ltd., «Open Letter to the Amateur Radio Community,» American Registry for Internet Numbers, [En línea]. Available: http://www.aredn.org/index.html.

[29] Internet Engineering Task Force, « Optimized Link State Routing Protocol (OLSR),» The Internet Society, 2003. [En línea]. Available: https://tools.ietf.org/html/rfc3626.

[30] Canonical Ltd., «Ubuntu - The leading operating system for PCs, IoT devices, servers and the cloud,» Canonical, 2018. [En línea]. Available: https://www.ubuntu.com/.

[31] OpenStack Project, «OpenStack - Cloud Operating System,» Rackspace Cloud Computing, 2018. [En línea]. Available: https://www.openstack.org/software/.

[32] Raspberry Pi Foundation, «Raspbian,» Raspberry Pi Foundation, 2018. [En línea]. Available: https://www.raspberrypi.org/downloads/raspbian/.

[33] LoRa Alliance™, «LoRaWAN™ Specification,» LoRa Alliance™, 2018. [En línea]. Available: https://www.lora-alliance.org/about-lorawan.

[34] LoRa Alliance™, «About LoRa Alliance™,» LoRa Alliance™, 2018. [En línea]. Available: https://www.lora-alliance.org/about-lora-alliance.

[35] Pycom Limited, «MicroPython,» Pycom, 2018. [En línea]. Available: https://pycom.io/support-2/faqs/what-is-micropython/.

[36] Python Software Foundation, «Python 3.0 Release,» Python Software Foundation, 2018. [En línea]. Available: https://www.python.org/download/releases/3.0/.

[37] Kickstarter, PBC, «Kickstarter - Haciendo realidad proyectos creativos.,» Kickstarter, 2018. [En línea]. Available: https://www.kickstarter.com/?lang=es.

[38] Pycom Limited, «LoRaWAN Nano-Gateway,» Pycom, 2018. [En línea]. Available: https://docs.pycom.io/chapter/tutorials/lora/lorawan-nano-gateway.html.

[39] GitHub, Inc., «Pycom Libraries Repository,» Pycom, 2018. [En línea]. Available: https://github.com/pycom/pycom-libraries.

[40] The Things Network, «The Things Network - Building a global Internet of thing network together,» The Things Network, 2018. [En línea]. Available: https://www.thethingsnetwork.org/.

[41] The Things Network, «The Things Network Stack,» The Things Network, 2018. [En línea]. Available: https://www.thethingsnetwork.org/tech-stack.

[42] Amazon Web Services, Inc., «AWS | Cloud Computing - Servicios de informática en la nube,» Amazon Web Services, 2018. [En línea]. Available: https://aws.amazon.com/es/.

[43] Microsoft Inc., «Microsoft Azure: plataforma y servicios de informática en la nube,» Microsoft, 2018. [En línea]. Available: https://azure.microsoft.com/es-es/.

[44] G. Hillar, MQTT Essentials - A Lightweight IoT Protocol, Packt Publishing, 2017.

GLOSARIO DE TÉRMINOS

Ad hoc: Ad hoc es un modo de funcionamiento de las redes inalámbricas que consiste en que todos los dispositivos se comunican entre sí sin la necesidad de un nodo central que coordine la comunicación y la conexión de nuevos dispositivos. Ad hoc es una manera fácil de comunicar dos dispositivos ya que solo es necesario dos tarjetas inalámbricas y configurarlas para que utilicen el mismo nombre de red.

Ancho de Banda ("*bandwith*"): En una conexión de red entre dos computadores, el ancho de banda es la cantidad de información o datos que se pueden transmitir por medio de un enlace de red dado un intervalo de tiempo. Las conexiones de red gigabit tienen un ancho de banda de 1 Gb/s, 125 MB/s o 119 MiB/s.

Anfitrión ("*host*"): Un host es un ordenador u otro dispositivo que está conectado a una red de ordenadores. Los hosts utilizan el protocolo IP para comunicarse con el resto de los dispositivos de la red y pueden ofrecer distintos servicios al resto de hosts.

ARM ("Advanced RISC Machine"): ARM es una arquitectura de la familia RISC ("reduced instruction set computing") para procesadores de computadores. Estos procesadores necesitan menos transistores que las arquitecturas CISC ("complex instruction set computing") lo que reduce el coste, el consumo energético y la disipación de calor.

CMS ("Content Management System" o "sistema de gestión de contenidos"): Un CMS es una aplicación informática que permite la creación y administración de contenidos de manera estructurada, principalmente en páginas web. La gestión se realiza por parte de los administradores, editores, participantes y demás usuarios del sitio. Cuenta con una interfaz en la que poder modificar el contenido o el diseño de manera individual.

Computación en la Nube ("cloud computing"): La computación en la nube define una serie de servicios de Internet que podemos alquilar para hacer uso de ellos durante el tiempo que necesitemos. De esta forma solo pagamos por los recursos que estemos consumiendo. Los servicios de cloud computing abarcan desde el simple procesamiento de datos hasta la creación de grandes redes virtuales complejas, pasando por servicios de almacenamiento de datos a medida. Gracias a la computación en la nube podemos desplegar todos nuestros servicios sin la necesidad de disponer de nuestro propio servidor web.

Corta Fuegos ("firewall"): Un firewall es un servicio de un sistema o de una red que está diseñado para bloquear el acceso no autorizado de hosts externos o aplicaciones no conocidas al sistema o a la red, permitiendo al mismo tiempo comunicaciones autorizadas o configuradas explícitamente para permitir su paso a través del firewall.

CPU ("Central Processing Unit" o "unidad de procesamiento central"): La CPU es la parte hardware de un ordenador que interpreta las instrucciones de un programa informático a través de operaciones aritméticas básicas, lógicas y de entrada/salida del sistema.

Datos a Gran Escala ("big data"): Big Data es el concepto que se refiere a los conjuntos de datos masivos que las aplicaciones informáticas tradicionales de procesamiento de datos no pueden abarcar por la cantidad y complejidad de los datos introducidos. Estos conjuntos son procesados por grandes computadores para encontrar patrones repetidos y poder sacar conclusiones a gran escala.

DHCP ("Dynamic Host Configuration Protocol" o "protocolo de configuración dinámica de host"): DHCP es un servicio de red que se encarga de asignar una dirección IP única a cada host que se conecta a la red y configura el resto de los parámetros para que ese host tenga conectividad con el resto de la red.

DNS ("Domain Name System" o "sistema de nombres de dominio"): DNS es un servicio de red que traduce los nombres de host en direcciones IP entendibles por las máquinas o viceversa, con el objetivo de localizar y direccionar estos equipos a nivel global.

Encaminador ("router"): Un router es un dispositivo de red que se encarga de transmitir paquetes entre diferentes redes con espacios de direccionamiento diferentes. Un router encamina un paquete basándose en su tabla de direcciones y además puede realizar otras tareas simultáneamente como fragmentado de paquetes o filtrar un tipo específico de tráfico.

Entrada/Salida de Propósito General ("GPIO" o "General Purpose Input/Output"): En un circuito integrado un pin GPIO entrega una señal digital que no está prefijada, es decir, que podemos configurar su comportamiento como entrada o como salida en tiempo de ejecución. La mayoría de estos pines no se usan para los componentes principales y son utilizados para añadir periféricos externos u otros componentes.

ERP ("Enterprise Resource Planning" o "sistema de planificación de recursos empresariales"): Los ERP son sistemas de información que integran y manejan muchos de los activos que intervienen en las operaciones de producción, los aspectos de distribución y la comunicación con clientes y proveedores de una compañía que se dedica a la producción de bienes y servicios.

GPS ("Global Positioning System" o "sistema de posicionamiento global"): GPS es un sistema controlado por satélite que proporciona geolocalización e información del tiempo a un receptor GPS situado en cualquier punto de la Tierra y con visión directa con el cielo. El sistema GPS pertenece al gobierno de E.E.U.U., mientras que la Unión Soviética dispone de su propio sistema denominado GLONASS, los chinos tienen su sistema BeiDou y la Unión Europea trabaja en otra alternativa llamada GALILEO.

GPU ("Graphics Processing Unit" o "unidad de procesamiento gráfica"): La GPU es un circuito electrónico diseñado para crear rápidamente imágenes en una memoria intermedia para luego ser mostradas en un monitor. Las GPUs actuales permiten obtener imágenes de alta resolución manteniendo un consumo adecuado para dispositivos móviles.

IPv4 ("Internet Protocol version 4"): La cuarta versión de este protocolo es uno de los pilares claves para que la conectividad a Internet sea posible. Fue desplegado en producción en 1983 y todavía es el protocolo más utilizado para transportar tráfico en Internet. Actualmente nos encontramos en la situación de agotamiento de direcciones IPv4 posibles.

IPv6 ("Internet Protocol version 6"): La sexta versión de este protocolo viene a sustituir a la cuarta versión y además de un "infinito" rango de direcciones proporciona otras muchas novedades para aumentar el rendimiento de Internet a muchos niveles.

NAT ("Network Address Translation"): El NAT es un método para enmascarar una dirección IP en otra modificando la cabecera del paquete IP mientras que este cruza un dispositivo enrutador. El NAT nos permite disponer un espacio de direccionamiento privado y a la vez puede comunicarse con otras redes a través de una o varias direcciones IP públicas.

Puerta de Enlace: ("gateway"): Una Gateway es un dispositivo de red que sirve como pasarela a otra red a la que transmitir los paquetes cuando la dirección destino no se encuentra dentro del rango de la red local y la Gateway actúa como ruta por defecto.

Red Privada Virtual ("VPN" o "virtual private network"): Una VPN puede ser utilizada para ampliar una red privada a través de una red pública o acceder a los recursos protegidos por un firewall desde un host externo. El tráfico de las aplicaciones que se conectan mediante una red VPN se encuentra encriptado y no puede ser visto desde fuera de ella. Esto proporciona un nivel más de seguridad y de privacidad a los usuarios que la utilicen.

Sistema en un Chip ("SoC" o "system on a chip"): Un SoC es un circuito integrado que contiene todos los componentes esenciales de un ordenador como CPU, RAM, puertos de entrada y salida, almacenamiento secundario e incluso una GPU o radios integradas para conectarse a redes inalámbricas. Los SoC son muy comunes de dispositivos móviles o aquellos que requieran disensiones reducidas y bajo consumo energético.

Soporte Lógico Inalterable ("firmware"): El firmware es el programa informático que establece la lógica de más bajo nivel que controla el comportamiento de los circuitos electrónicos de un dispositivo de cualquier tipo. El firmware define el comportamiento más básico de un sistema, ya que es el software que se comunica directamente con el hardware.

Tecnología 5G: La tecnología 5G es la sucesora a la actual tecnología de conexión móvil 4G. 5G promete ligeras mejoras en la velocidad de descarga al contar con más frecuencias de funcionamiento y podrá alcanzar velocidades de hasta 1Gbit/s cuando combinamos todas las frecuencias anteriores (con 4G podemos alcanzar hasta 300Mb/s).

Telefonía IP ("VoIP" o "voice over IP"): La telefonía IP es un conjunto de tecnologías que permiten ofrecer servicios de comunicación por voz y sesiones multimedia (videoconferencias, presentaciones) sobre una red IP como una red corporativa o Internet.

V2V ("Vehicle-to-vehicle"): La comunicación vehículo a vehículo consiste en formar redes ad hoc durante un trayecto para intercambiar información útil entre dos o más vehículos. Esta comunicación emplea la banda sin licencia de 5.9 GHz.

Zigbee y Z-wave o Sub-GHz: Estos conceptos se refieren a protocolos de comunicación que se utilizan para la creación de redes de área personal en entornos domésticos, como puede ser las comunicaciones entre bombillas inteligentes, electrodomésticos o sensores y actuadores. Zigbee emplea la banda de 2.4Ghz mientras que Z-wave utiliza la frecuencia de 900 MHz, como consecuencia esta última estará limitada en términos de ancho de banda pero ofrecerá una cobertura más extensa que Zigbee.

CONTENIDO DEL CD

En el CD que acompaña a la memoria podemos encontrar los siguientes recursos:

- Memoria del trabajo en formato PDF dentro del directorio Memoria.
- Hoja de especificaciones de los módulos LoPy, Pysense, Raspberry Pi 3B, Beagle-Bone Black y el módulo Wi-Pi dentro del directorio Specsheets.
- Documentación y manuales que se han utilizado para la realización del TFM. Los cuales los podemos encontrar dentro del directorio Bibliografía.
- El proyecto HSMM-Pi descargado desde GitHub dentro del directorio Proyecto.
- Los ficheros fuente personalizados para este trabajo de la Gateway LoRa y el nodo cliente dentro del directorio Proyecto.
- La aplicación utilizada para actualizar el firmware de las placas LoPy y Pysense en el directorio Pycom Update.
- La imagen de instalación de Ubuntu Linux 14.04.3 para la BeagleBone Black dentro del directorio Instalación.
- La imagen de NOOBS para la instalación de Raspbian Linux en las Raspberry Pi 3B dentro del directorio Instalación.
- Aplicación iPerf3 versión 3.1.3 para Windows y MAC en el directorio iPerf.