



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Máster en Ingeniería de Computadores y Redes
Trabajo Fin de Máster

Redes de Área Extensa para aplicaciones de IoT: modelado de comunicaciones Sigfox

Autor: *Pablo Pardal Garcés*

Director(es): *José Carlos Campelo Rivadulla*

Alberto Miguel Bonastre Pina

14 de Julio de 2017

Resumen

Hoy en día vivimos en un mundo conectado en el que el Internet de las Cosas (IoT, “*Internet of Things*”) está cobrando cada vez más importancia. Este concepto hace referencia a la interconexión de cualquier objeto con Internet a través de diferentes tecnologías de comunicación. Para ello los dispositivos IoT pueden emplear redes de datos ya existentes, como pueden ser LTE, WiFi o Bluetooth, si bien surgen nuevas tecnologías desarrolladas para el sector IoT, entre las que pueden citarse las denominadas “*Low Power Wide Area Network*” (LPWAN), como LoRa, Sigfox o RPMA.

Al estar concebidas específicamente para este entorno IoT, estas nuevas tecnologías, que pretenden convertirse en futuros estándares, ofrecen unas prestaciones muy interesantes. A la hora de elegir la tecnología de comunicación más apropiada para cada tipo de aplicación es importante conocer sus capacidades, así como las ventajas que ofrecen unas respecto de otras. Por este motivo, en este trabajo se ha realizado un estudio teórico de las principales tecnologías de comunicación LPWAN.

Las conclusiones de este estudio teórico apuntan que las características de Sigfox hacen que sea una de las tecnologías más prometedoras del sector y está presente en muchos dispositivos. Es por ello que, con el fin de poder verificar y analizar detalladamente estas características, se ha implementado un modelo NS-3 de esta tecnología de comunicación. Con este modelo es posible estudiar muchas características en diferentes entornos, como la cobertura, el consumo de potencia o la tasa de errores.

La intención última de este trabajo, que se pretende reflejar en forma de Tesis Doctoral, es apoyarse en este modelo para proponer nuevos protocolos que mejoren las prestaciones de Sigfox y permitan superar sus limitaciones.

Resum

Hui en dia vivim en un món connectat en el que l'Internet de les Coses (IoT, "Internet of Things") està cobrant cada vegada més importància. Aquest concepte fa referència a la interconnexió de qualsevol objecte amb Internet a través de diferents tecnologies de comunicació. Per a això els dispositius IoT poden emprar xarxes de dades ja existents, com poden ser LTE, WiFi o Bluetooth, si bé sorgixen noves tecnologies desenrotllades per al sector IoT, entre les que poden citar-se les denominades "Low Power Wide Àrea Network" (LPWAN) , com LoRa, Sigfox o RPMA.

A l'estar concebudes específicament per a este entorn IoT, estes noves tecnologies, que pretenen convertir-se en futurs estàndards, oferixen unes prestacions molt interessants. A l'hora de triar la tecnologia de comunicació més apropiada per a cada tipus d'aplicació és important conèixer les seues capacitats, així com els avantatges que oferixen unes respecte d'altres. Per este motiu, en este treball s'ha realitzat un estudi teòric de les principals tecnologies de comunicació LPWAN.

Les conclusions d'aquest estudi teòric apunten que les característiques de Sigfox fan que siga una de les tecnologies més prometedores del sector i està present en molts dispositius. És per això que, a fi de poder verificar i analitzar detalladament estes característiques, s'ha implementat un model NS-3 d'aquesta tecnologia de comunicació. Amb este model és possible estudiar moltes característiques en diferents entorns, com la cobertura, el consum de potència o la taxa d'errors.

La intenció última d'aquest treball, que es pretén reflectir en forma de Tesi Doctoral, és recolzar-se en este model per a proposar nous protocols que milloren les prestacions de Sigfox i permeten superar les seues limitacions.

Abstract

Nowadays we live in a communicated world in which the concept of Internet of Things (IoT) is becoming more important. This concept refers to the digital interconnection of any object with internet through different technologies. Technologies that can be the traditional, such as LTE, WiFi or Bluetooth, or the recently developed for the IoT world, called Low Power Wide Area Networks (LPWAN), such as LoRa, Sigfox or RPMA among others.

These new technologies, which aim to become future standards, offer many interesting capabilities. When choosing the most appropriate communication technology for each type of application is important to know their capabilities, as well as the advantages they offer respect to each other. Due to this reason, a theoretical study of the main LPWAN technologies has been carried out.

The conclusions of this theoretical study suggest that the characteristics of Sigfox make it one of the most promising technologies in the sector and is present in many devices. It is for this reason that, in order to verify and analyze these characteristics in detail, a NS-3 model of this communication technology has been implemented. With this model is possible to study many characteristics in different environments, such as coverage, power consumption or error rate.

The final intention of this work, which pretends to be a future Doctoral Thesis, is to rely on this model to propose new protocols that improve Sigfox beyond its limitations.

Índice general

| | |
|--|----|
| Capítulo 1. Introducción | 11 |
| 1.1 Motivación | 11 |
| 1.2 Objetivos | 11 |
| 1.3 Metodología | 12 |
| 1.4 Organización del TFM | 12 |
| Capítulo 2. Internet de las Cosas..... | 13 |
| 2.1 Conceptos básicos | 14 |
| 2.2 Arquitectura IoT..... | 16 |
| 2.2.1 Arquitectura de tres niveles con objetos conectados sin protocolo IP | 17 |
| 2.2.2 Arquitectura de dos niveles con objetos conectados con protocolo IP | 18 |
| 2.2.3 Arquitectura de dos niveles con objetos conectados sin protocolo IP..... | 19 |
| 2.3 Tecnologías IoT..... | 19 |
| 2.3.1 Tecnología en objeto conectado..... | 20 |
| 2.3.2 Tecnologías de comunicación | 20 |
| 2.3.3 Tecnologías aplicativas de red | 21 |
| 2.3.4 Tecnologías Cloud y Big Data | 21 |
| 2.3.5 Tecnologías de aplicación IoT | 21 |
| 2.4 Aplicaciones | 22 |
| 2.4.1 Ciudades inteligentes..... | 22 |
| 2.4.2 Domótica | 23 |
| 2.4.3 Industria..... | 23 |
| 2.4.4 Salud..... | 24 |
| 2.4.5 Eficiencia energética | 24 |
| 2.4.6 Sector agrícola..... | 24 |
| 2.4.7 Turismo | 25 |
| Capítulo 3. Tecnologías de comunicación IoT | 26 |
| 3.1 ETSI LTN..... | 28 |
| 3.2 Sigfox | 29 |
| 3.3 LoRa..... | 29 |
| 3.4 RPMA | 30 |
| 3.5 Weightless | 30 |
| 3.5.1 Weightless-N..... | 31 |
| 3.5.2 Weightless-P..... | 31 |
| 3.5.3 Weightless-W | 31 |
| 3.6 NB-Fi..... | 32 |

| | | |
|-------------|--|----|
| 3.7 | Grupo 3GPPP | 32 |
| 3.8 | DASH7 | 33 |
| 3.9 | IEEE 802.11ah | 33 |
| 3.10 | Comparación | 34 |
| 3.11 | Caso real..... | 37 |
| Capítulo 4. | ETSI - Low Throughput Network (LTN)..... | 40 |
| 4.1 | Arquitectura..... | 40 |
| 4.2 | Interfaz A..... | 42 |
| 4.2.1 | UNB | 42 |
| 4.2.2 | OSSS | 46 |
| Capítulo 5. | Simuladores de redes de telecomunicación..... | 49 |
| 5.1 | NCTUns | 49 |
| 5.2 | Kiva..... | 50 |
| 5.3 | COMNET III..... | 50 |
| 5.4 | OMNeT++..... | 51 |
| 5.5 | OPNET MODELER..... | 51 |
| 5.6 | NS-2 | 52 |
| 5.7 | NS-3 | 52 |
| 5.8 | Comparativa | 52 |
| Capítulo 6. | Network Simulator 3 (NS-3) | 54 |
| 6.1 | Características | 54 |
| 6.2 | Comparación con NS-2 | 54 |
| 6.3 | Organización del software..... | 55 |
| 6.3.1 | Orientación a objetos..... | 56 |
| 6.3.2 | Estructura de un script NS-3 | 57 |
| Capítulo 7. | Modelo NS-3 de Sigfox..... | 60 |
| 7.1 | Estructura | 60 |
| 7.2 | Módulos..... | 61 |
| 7.2.1 | Capa PHY..... | 61 |
| 7.2.2 | Capa MAC..... | 65 |
| Capítulo 8. | Estudios realizados..... | 68 |
| 8.1 | Estudio de alcance | 68 |
| 8.1.1 | Entorno rural | 70 |
| 8.1.2 | Entorno suburbano | 71 |
| 8.1.3 | Entorno urbano..... | 72 |
| 8.2 | Estudio de consumo de potencia | 72 |
| 8.3 | Estudio del método de acceso al medio (RFTDMA) | 73 |

| | |
|--|----|
| Capítulo 9. Conclusiones y trabajo futuro..... | 75 |
| Bibliografía..... | 77 |
| Anexo - Acrónimos..... | 79 |

Índice de figuras

| | |
|--|----|
| Figura 1 - Objetos IoT | 13 |
| Figura 2 - Componentes IoT | 16 |
| Figura 3 - Arquitectura de tres niveles sin protocolo IP | 17 |
| Figura 4 - Arquitectura de dos niveles con protocolo IP | 18 |
| Figura 5 - Arquitectura de dos niveles sin protocolo IP | 19 |
| Figura 6 - Tecnologías IoT | 20 |
| Figura 7 - Ejemplo aplicaciones | 22 |
| Figura 8 - Domótica..... | 23 |
| Figura 9 - Medición inteligente | 24 |
| Figura 10 - Situación LPWAN | 26 |
| Figura 11 - Comparativa de redes..... | 27 |
| Figura 12 - Situación del mercado IoT | 28 |
| Figura 13 - Comparativa Weightless | 30 |
| Figura 14 - Comparativa 3GPP..... | 33 |
| Figura 15 - Comparación alcances..... | 36 |
| Figura 16 - Comparativa penetración | 37 |
| Figura 17 - Arquitectura LTN..... | 40 |
| Figura 18 - UNB vs OSSS | 42 |
| Figura 19 - Ejemplo RFTDMA | 43 |
| Figura 20 - Trama UNB MAC de subida..... | 44 |
| Figura 21 - Trama UNB MAC de bajada..... | 45 |
| Figura 22 - Sincronización UNB | 45 |
| Figura 23 - Trama OSSS física..... | 47 |
| Figura 24 - Trama OSSS MAC..... | 47 |
| Figura 25 - Sistema de eventos discretos | 55 |
| Figura 26 - Organización software NS-3 | 55 |
| Figura 27 - Directorio /src | 56 |
| Figura 28 - Modelo de objetos | 57 |
| Figura 29 – Diagrama modelo Sigfox NS-3 | 60 |
| Figura 30 - Máquina de estados PHY | 62 |
| Figura 31 - Espectro del enlace..... | 63 |
| Figura 32 - Dependencias Sigfox-spectrum..... | 63 |
| Figura 33 - Dependencias Sigfox-lqi | 63 |
| Figura 34 - Dependencias Sigfox-error-model | 64 |
| Figura 35 - Dependencias Sigfox-energy-model | 65 |
| Figura 36 - Estados capa MAC..... | 66 |
| Figura 37 - Dependencias MAC header..... | 66 |
| Figura 38 – Dependencias MAC trailer | 67 |
| Figura 39 - Perdidas básicas Okumura-Hata | 69 |
| Figura 40 - Factor de corrección rural | 69 |
| Figura 41 - Factor de corrección suburbano | 69 |
| Figura 42 - Factor de corrección ciudades pequeñas | 69 |
| Figura 43 - Factor de corrección ciudades grandes..... | 69 |
| Figura 44 - Escenario rural | 70 |
| Figura 45 - Escenario suburbano | 71 |
| Figura 46 - Escenario urbano..... | 72 |
| Figura 47 - Consumo de potencia | 73 |

Índice de tablas

| | |
|---|----|
| Tabla 1 - Comparación LPWANs..... | 35 |
| Tabla 2 - Prestaciones LoRa (TCP) | 38 |
| Tabla 3 - Prestaciones LoRa (UDP)..... | 39 |

Capítulo 1. Introducción

En los últimos años ha surgido el concepto del Internet de las Cosas (IoT, “Internet of Things”) como una idea a través de la cual el modelo de Internet se convertiría en un modelo enfocado a una conectividad total, donde todos los objetos que podamos pensar podrían estar conectados a la red.

Este modelo, se ha convertido hoy en día en uno de los temas más importantes de investigación y de negocio en las empresas tecnológicas. Este crecimiento radical en el número de dispositivos conectados a Internet en estos últimos diez años ha llevado a que actualmente sea mayor el número de dispositivos conectados que el número de habitantes en el planeta.

De esta forma se consolida el hecho de que la mayor parte de la información de la red es generada por objetos conectados en múltiples ámbitos de aplicación, como la domótica, las ciudades inteligentes, los automóviles inteligentes o la telemedicina, entre muchos otros.

La larga lista de ventajas en distintos sectores que conlleva esta revolución justifica el esfuerzo en la mejora de las redes de comunicaciones. Es necesario avanzar, entre otros, en aspectos como la seguridad de las redes y los dispositivos, el reparto del ancho de banda, la instalación de la infraestructura de soporte, el consumo de los dispositivos... Es por ello que las alternativas de soporte a redes IoT es un campo de investigación muy interesante.

1.1 Motivación

La importancia que está cobrando en los últimos años el sector IoT y las redes de sensores ha puesto de manifiesto las carencias de las redes de comunicaciones inalámbricas existentes. La disparidad de aplicaciones no permite hablar de una tecnología que aporte una solución general a todas las aplicaciones posibles. Sin embargo, es posible identificar dos tendencias principales. Por un lado, algunas tecnologías, orientadas sobre todo a comunicaciones entre computadores, proporcionan un elevado ancho de banda a cortas distancias. Por el otro, otras sacrifican el ancho de banda para proporcionar mayores coberturas. Otro factor importante es el coste energético de las comunicaciones, especialmente relevante con dispositivos móviles que dependen de baterías para su funcionamiento. Por ello, se convierte en una necesidad estudiar las características y las prestaciones de las tecnologías de comunicación más adecuadas para aplicaciones en el entorno IoT. Este trabajo se centra en Sigfox, una de las más implementadas y con un futuro muy prometedor. En esta línea, se considera muy interesante la implementación de un modelo de simulación de esta tecnología, como un primer paso de un estudio detallado que conduciría a propuestas de mejora y nuevas soluciones.

1.2 Objetivos

El objetivo principal de este trabajo final de máster es realizar un estudio de las distintas tecnologías de comunicación del sector IoT y diseñar un modelo de simulación para una red basada en la tecnología Sigfox.

Una vez realizado el modelo, se someterá a una campaña de simulación exhaustiva con el fin de estudiar sus prestaciones en parámetros tales como la cobertura, el consumo de potencia, la tasa de errores o la calidad del enlace.

Esta información se considera de gran valor, puesto que permite realizar comparaciones y estudios con otras tecnologías de diferentes características. Incluso permitiría detectar puntos débiles en la tecnología y proponer mejoras que cubran las carencias de Sigfox o aconsejen otras tecnologías.

1.3 Metodología

La elaboración del estado del arte concerniente a las tecnologías de comunicación dentro del sector IoT se ha basado en el estudio de diferentes trabajos de investigación y en el análisis de la información aportada por los grupos y compañías que regulan las distintas tecnologías de comunicación estudiadas. Del mismo modo, los puntos que referencian a los simuladores de redes de comunicación, y en concreto a NS-3, han sido elaborados con el mismo criterio.

Para programar el modelo de Sigfox se ha utilizado como guía otros ya implementados en el simulador NS-3, como modelos preexistentes del estándar IEEE 802.15.4 (ZigBee) o IEEE 802.11 (WiFi).

Los datos obtenidos en las simulaciones han sido comparados y contrastados con las especificaciones declaradas por Sigfox, obteniendo una elevada correspondencia entre ambos datos. Por lo tanto, asumiendo que las especificaciones de Sigfox son correctas, se considera el modelo como representativo.

1.4 Organización del TFM

Una vez descrita la introducción a la cual pertenece el presente apartado, donde se exponen motivación, objetivos y metodología, a continuación, se explicará cómo se ha estructurado el resto del proyecto.

En el punto 2 se realiza una descripción detallada del sector IoT, qué es exactamente, cómo se estructura, qué tecnologías pueden aplicarse y en qué ámbitos.

Después, en el *punto 3*, se realiza un estudio de las distintas tecnologías de comunicación que han aparecido en los últimos años. De estas, en el *punto 4*, se hará énfasis en el estándar ETSI LTN, en el cual se basa la tecnología Sigfox.

Seguidamente, en el *punto 5*, se elabora un estudio de los distintos simuladores de redes de comunicación, para más adelante, en el *punto 6*, describir en profundidad el simulador utilizado: NS-3.

Como paso final del trabajo, en el *punto 7* se comenta como se ha programado el modelo NS-3 de Sigfox para en el *punto 8* presentar las diferentes simulaciones realizadas y los resultados obtenidos.

Finalmente, las conclusiones al trabajo realizado y los trabajos futuros previstos, se han expuesto en los *puntos 9 y 10* respectivamente.

Capítulo 2. Internet de las Cosas

El concepto “*Internet de las Cosas*”, o “*Internet of Things*” (IoT), tiene una definición bastante amplia y difusa, pero puede resumirse como un paradigma tecnológico que define la dotación de conectividad a internet a cualquier objeto sobre el que se pueda medir parámetros físicos o actuar, así como las aplicaciones y tratamiento de datos inteligentes relativos a los mismos.



Figura 1 - Objetos IoT

Este término de “*Internet de las cosas*”, aunque era comúnmente utilizado en círculos de investigación desde 1999, es bastante reciente. Fue usado de forma pública por primera vez en 2009 por el profesor Kevin Ashton en el *RFID journal* (Ashton, 2009) y desde esa publicación su uso ha ido creciendo de forma exponencial.

Actualmente el concepto está ampliamente utilizado en todo tipo de artículos tecnológicos, ya sea de forma directa o tangencial. Para el entender el origen del IoT se han de analizar las evoluciones de las distintas tecnologías.

El primer objeto conectado del que se tiene constancia fue la instalación de un dispositivo de información meteorológica y profundidad de la nieve en la cima del *Mont Blanc* por científicos franceses en el año 1874. A través de una comunicación de radio de onda corta la información era transmitida a París. En adelante, durante el siglo XX, hubo un crecimiento en la telemetría impulsado por la evolución de las tecnologías de telecomunicación.

En la década de los 60 y los 70 se crearon los primeros protocolos de comunicación de uso militar y académico en los Estados Unidos, que definirían la red ARPANET, la cual es considerada la base del actual internet.

Durante esta década de los 70 y los posteriores 80 se dio paso a la creación de redes heterogéneas debido a la falta de tecnología para comunicaciones a larga distancia, siendo estas totalmente incompatibles entre sí.

No fue hasta mediados de los 90 cuando se creó un protocolo de comunicación que permitía interconectar estas redes: el protocolo TCP/IP, dando origen al inicio de lo que actualmente conocemos como Internet. De esta forma se dio origen a una gran variedad de nuevos modelos sociales y de negocio.

Ante la rápida popularización de Internet, la idea de conectar objetos empezó a coger fuerza. En 1990 el ingeniero John Romkey creó el primer objeto conectado a Internet a través del protocolo TCP/IP: una tostadora que podía apagarse y encenderse remotamente.

A pesar de la revolución que este concepto de objetos conectados suponía, la idea no cogió fuerza debido al nivel tecnológico del momento. Las comunicaciones a Internet eran principalmente cableadas y el coste del hardware era elevado, dando lugar a limitaciones en la conectividad y la creación de redes y dispositivos de medición.

Con el desarrollo y la popularización de las redes inalámbricas, como WiFi o redes celulares, durante el siglo XXI se presencia una explosión en el crecimiento del IoT. Este crecimiento se ha sucedido con la aparición de las Redes de Sensores (“*Wireless Sensor Networks*”, WSN) o las conexiones Máquina a Máquina (“*Machine to Machine*”, M2M), las cuales han dado lugar al actual “*Internet de las Cosas*”. A diferencia de los demás, IoT abarca conceptos en cuanto a protocolos, estándares, consorcios, redes, apertura de interfaces, tecnologías de comunicación y entornos de desarrollo, dándole a su vez unas mayores expectativas en el futuro.

El estudio sobre “*Internet de las Cosas*”, objetivo de este punto, se ha estructurado en apartados que tratan sobre los conceptos básicos, la arquitectura, las tecnologías y los estándares y los consorcios del IoT.

2.1 Conceptos básicos

Aunque las bases del IoT se están desarrollando sobre ideas y tecnologías ya existentes y consolidadas, su gestión conlleva un gran problema debido a la gran variedad en cuanto a casos de aplicación, muy diferentes unos de otros. Es por ello que es conveniente que se especifiquen ciertos requerimientos y máximas tecnológicas a modo de conceptos básicos:

- ***Los objetos conectados pueden ser inteligentes, pero no es obligatorio:***

Debido a la reducción de costes en cuanto a la producción de microprocesadores de mayor capacidad y con un consumo reducido de energía, se ha provocado la aparición de pequeños dispositivos capaces de obtener y procesar información. En ciertos casos de uso es necesario que el dispositivo de medición realice a su vez un análisis de los datos y un pretratamiento de la información antes de ser transmitida. Pero IoT no obliga a que se realice de esta forma. En muchos casos el tratamiento de la información se realiza a nivel de *cloud*, de plataforma o de aplicación, obteniendo así un consumo mínimo en el dispositivo. Cada caso de uso determinará el nivel de inteligencia del objeto.

- ***Es conveniente que exista conectividad IP punto a punto, aunque en determinados casos no sea aplicable:***

IoT está basado en su conectividad a Internet, cuya base fundamental es el protocolo IP. El caso ideal sería una conectividad total sobre IP, ya que permitiría la conexión a los dispositivos sin necesidad de usar repetidores o pasarelas.

A nivel físico, las conexiones cableadas presentan evidentes inconvenientes en cuanto a movilidad, siendo aplicables en pocos casos. Para dotar de conectividad inalámbrica a un objeto hay que tener en cuenta el ámbito de aplicación en el que nos encontramos y recurrir a la tecnología de conexión más apropiada. Entre estas tecnologías tenemos:

- *WiFi*: consumo energético alto y coste del dispositivo reducido.
- *Redes celulares (2G, 3G o 4G)*: consumo energético alto y un coste de conectividad considerable.
- *6LowPAN*: tecnología de radio de uso reducido basado en IPv6.
- *Redes mercado IoT*: redes de radio de bajo consumo desarrolladas por el mercado, denominadas LPWA (“*Low Power Wide Area*”). Algunos ejemplos los tenemos en Sigfox, Lora o Weighless.

Finalmente, de forma independiente al nivel físico, los dispositivos que implementan el protocolo IP deben enfrentarse a factores limitantes en cuanto al coste del dispositivo, su consumo energético y el coste de la conectividad

- ***La conectividad de red ha de ser abierta y estándar:***

Actualmente las propuestas de implementación de IoT presentan una gran heterogeneidad debida, entre otros factores empresariales, a la variedad de usos y requerimientos de cada aplicación. Este hecho hace que no sea trivial la estandarización, como ocurrió en su momento con la telefonía móvil en la cual el requerimiento único era hablar.

Es por ello que no siempre es posible el uso de las redes estándar de comunicación, lo que lleva al desarrollo y creación de nuevas tecnologías para cubrir las necesidades que las ya implantadas, como las redes de telefonía móvil, no pueden abarcar. De esta forma nacen como un esfuerzo de estandarización las *LPWA*.

Finalmente, resulta fundamental estructurar mecanismos de interoperabilidad que permitan la coexistencia, no solo de diversas tecnologías, sino de diferentes filosofías a la hora de aplicar el concepto de IoT.

- ***El “cloud” y las técnicas de “Big Data” son necesarias en IoT cuando el volumen del despliegue justifique su uso:***

Habitualmente, en un ecosistema IoT de gran tamaño la ingente cantidad de información obtenida requiere técnicas de *Big Data* como elemento clave en despliegues. Puesto que la mera interacción con los objetos conectados no siempre es suficiente, un correcto manejo de la información obtenida por estos dispositivos puede aportar un valor añadido a los datos y a su uso.

2.2 Arquitectura IoT

La arquitectura del “*Internet de las Cosas*” se puede estructurar conforme podemos ver en la *figura 2*, teniendo como áreas principales:

- Objetos conectados
- Tecnologías de red
- Protocolos de comunicación
- Plataforma IoT de tratamiento de datos
- Aplicaciones de usuario

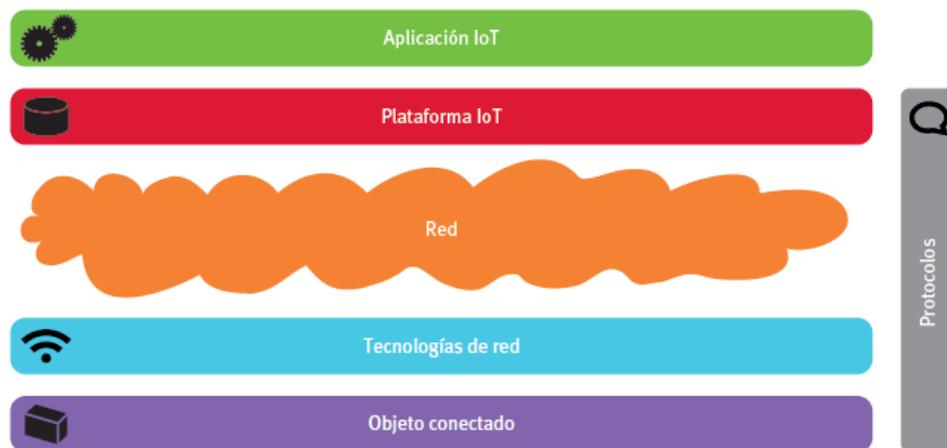


Figura 2 - Componentes IoT

Muchas de las propuestas de implementación de IoT se basan en los componentes de la *figura 2*. En la mayoría de los casos son aplicaciones de uso específico que presentan diferencias entre ellas a nivel de objeto conectado e integración con la nube o plataforma IoT. De este modo, se diferencian tres tipos fundamentales de arquitectura según la aplicación:

- Arquitectura de tres niveles con objetos conectados sin protocolo IP
- Arquitectura de dos niveles con objetos conectados con protocolo IP
- Arquitectura de dos niveles con objetos conectados sin protocolo IP

El primer tipo de arquitectura se corresponde con los despliegues con radios de baja potencia que usan repetidores o pasarelas para conectarse a una red IP. El segundo tipo incorpora tecnología con conectividad directa sobre IP, como pueden ser WiFi o 2G. La tercera arquitectura usa nuevos protocolos de red específicos para IoT con su propia red de comunicación, sin usar el protocolo IP.

Las tres arquitecturas tienen un nivel inferior común en el que se entiende la presencia de un dispositivo físico conectado. Pueden ir desde algo sencillo como sensores, acopladores, o interruptores, a dispositivos más complejos con una mayor capacidad de proceso que les dota de una cierta inteligencia.

A partir de este nivel, las diferentes formas de interconexión convierten al objeto en un dispositivo IoT, dependiendo de la aplicación, la infraestructura disponible, el coste, decisiones de los fabricantes o, incluso, factores políticos.

2.2.1 Arquitectura de tres niveles con objetos conectados sin protocolo IP

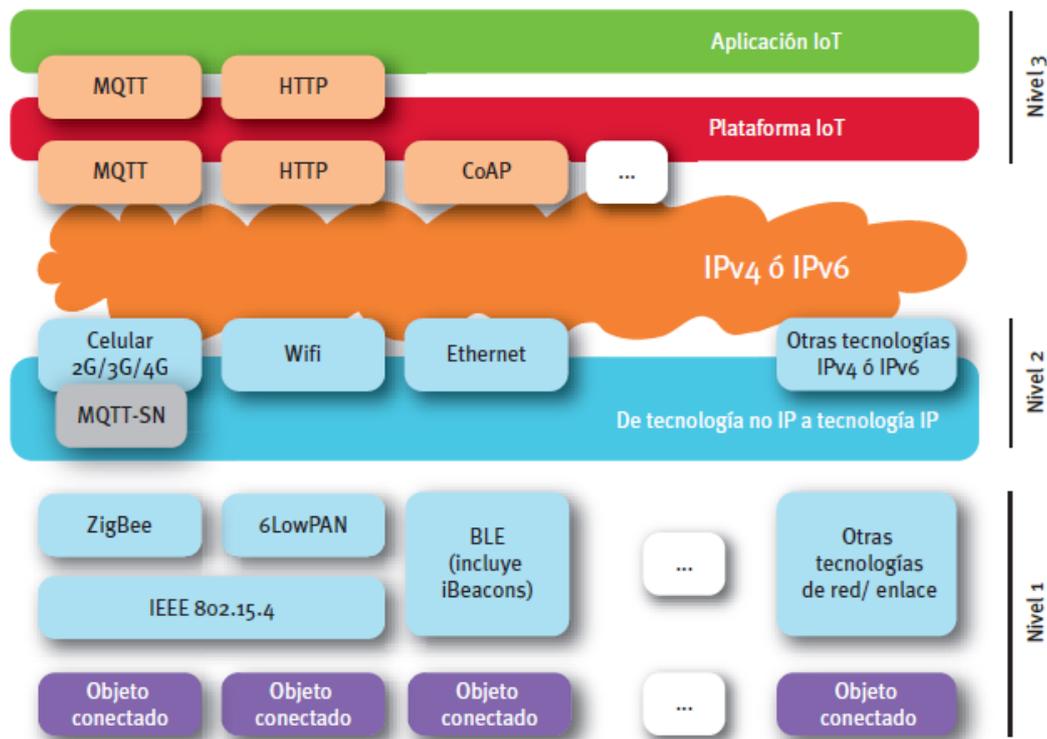


Figura 3 - Arquitectura de tres niveles sin protocolo IP

Este tipo de arquitectura ha sido la más utilizada desde hace años en sistemas que requieren la conectividad de un gran número de dispositivos de bajo coste y baja capacidad como contadores o sensores alimentados por batería. En este caso, el consumo energético que provoca el uso de la pila de protocolo IP es elevado, por lo que se opta por el uso de pasarelas o repetidores para salir a una red IP.

El primer nivel de conectividad tiene dos posibles planteamientos:

- Conexión directa punto a punto a través de una pasarela IP. Como IEEE 802.15.4 o “Bluetooth Low Energy” (BLE)
- Conectividad enrutada o mallada sin uso de IP hacia una pasarela IP. Como ZigBee, Z-WAVE o Wireless-HART, entre otros

Una vez conectado a la pasarela, estas serán las encargadas de ofrecer conectividad IP sobre otras tecnologías típicas de este tipo de conexión, como IEEE 802.11, IEEE 802.3, GPRS o basadas en Fibra óptica (FTTx) entre otras. Una vez conseguida conectividad IP se establecen conexiones UDP o TCP.

La mayoría de las arquitecturas están basadas en comunicaciones que ofrecen un servicio web sobre HTTP usando como encapsuladores de datos formatos como XML. En los últimos años están ganando importancia protocolos propios de IoT como MQTT o CoAP.

Por último, el tercer nivel de la arquitectura se encarga de realizar la conexión entre la nube o la plataforma IoT, dependiendo de la aplicación. Sobre esta capa, se ofrece un servicio para el desarrollo de aplicaciones para la interacción con el sistema, ya sea para control de los dispositivos o presentación de datos.

2.2.2 Arquitectura de dos niveles con objetos conectados con protocolo IP

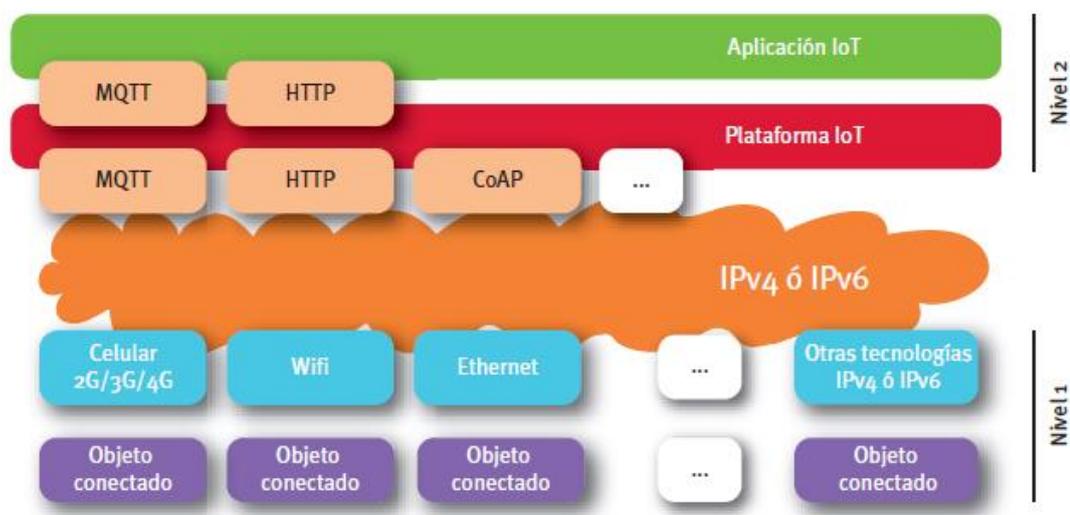


Figura 4 - Arquitectura de dos niveles con protocolo IP

Este tipo de arquitectura IoT está presente en objetos equipados con módulos de conexión sobre IP (WiFi, 2G, 3G o 4G) haciendo que no sea necesario el uso de pasarelas. Su uso se justifica en aplicaciones donde los objetos no tienen cobertura con ninguna red o en objetos con una alta autonomía de energía, ya sea porque están equipados con baterías de gran capacidad o porque realizan un reducido número de transmisiones diarias, reduciéndose así el consumo.

De esta forma, los dispositivos son capaces de conectarse directamente con el siguiente nivel, con la plataforma IoT o con la nube. Esta independencia en cuanto a la conectividad obliga a que los objetos sean más complejos, integrando un hardware más potente con mayor capacidad de proceso y memoria. En este caso, los protocolos sobre IP y TCP o UDP serán MQTT, CoAP o HTTP con servicios web RESTful.

Aunque el mercado con este tipo de tecnologías no está evolucionado, es una solución que está creciendo y empieza a tener cada vez más presencia en aplicaciones específicas.

2.2.3 Arquitectura de dos niveles con objetos conectados sin protocolo IP

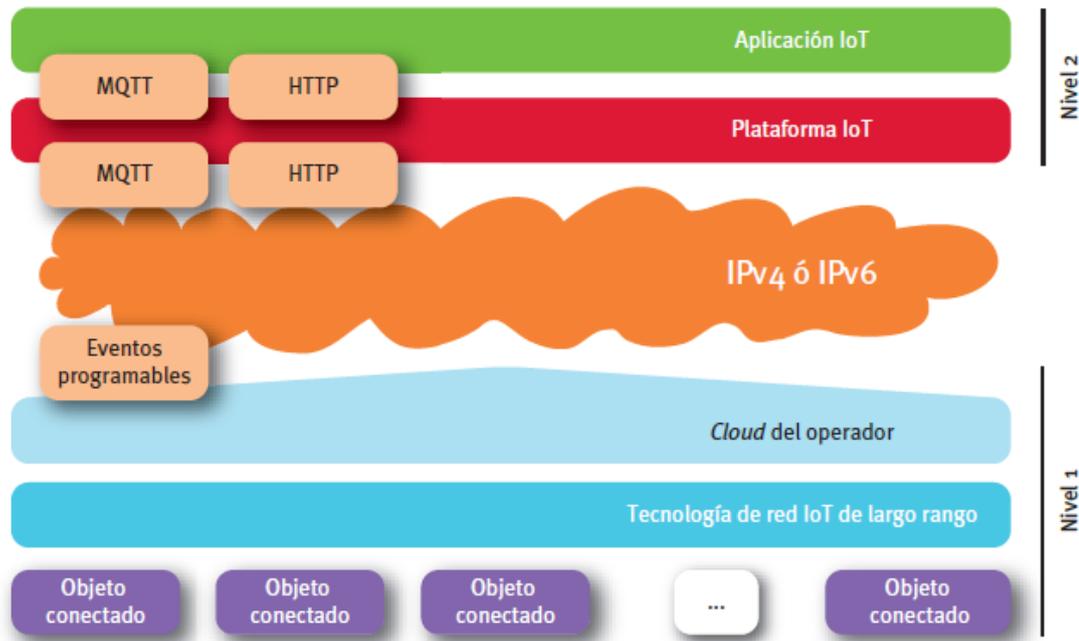


Figura 5 - Arquitectura de dos niveles sin protocolo IP

En este tipo de arquitectura tenemos una estructura de dos niveles como en la figura 5, similar a los despliegues que encontramos en redes celulares.

Con la aparición del mundo del IoT los protocolos existentes han ido evolucionando para adaptarse a este nuevo concepto y, junto al desarrollo de nuevas soluciones técnicas, simplificar los despliegues, ganar cobertura de red y reducir el consumo energético y el coste de los objetos. Estas tecnologías se basan en protocolos propios no dependientes de IP, siendo capaces de ofrecer una conexión directa o casi directa entre el dispositivo y la plataforma.

De la mano de esta arquitectura surgen nuevos protocolos y operadores IoT que ofrecen una conexión directa. Entre ellos podemos encontrar soluciones como las de Sigfox, LoRa o Weighless, las cuales ofrecen objetos conectados de forma directa a la red, capacidad de crear dispositivos de bajo coste y larga duración de batería y un servicio de operador de comunicaciones, evitando que el usuario tenga que usar pasarelas o repetidores.

2.3 Tecnologías IoT

Las distintas arquitecturas presentes en el mundo IoT dejan componentes abiertos a su integración con distintas tecnologías en función de la aplicación, las redes disponibles, el consumo energético, los costes de producción o mantenimiento y las implicaciones industriales o políticas. De esta forma, se puede hacer una clasificación de las distintas tecnologías del IoT.



Figura 6 - Tecnologías IoT

2.3.1 Tecnología en objeto conectado

Todo objeto conectado tiene su base en plataformas hardware y software, siendo necesaria una capacidad de proceso y una programación de la misma. Los productos IoT actuales basan principalmente su arquitectura de microcontroladores en núcleos ARM, PIC, AVR, 808X o los ARM-Cortex M. Sobre estos hay una gran variedad de sistemas operativos, de los cuales la mayoría han surgido como propuestas específicas para el sector (TinyOS, RIOT OS, Contiki o mbed OS).

Plataformas de hardware abierto como Arduino y Raspberry Pi han tenido un importante papel en cuanto a la evolución del IoT y los objetos. Ambos han servido como plataforma de desarrollo para la concepción de nuevas ideas y conceptos, pero su aportación en entornos exigentes es limitada al no estar diseñados para este fin. De este modo, aparecen diferentes plataformas hardware para desarrollo IoT, ya sean propietarias o abiertas, con intención de copar el mercado, como puede ser el caso de mbed de ARM.

2.3.2 Tecnologías de comunicación

La conectividad en el mundo IoT no tiene una única forma de implementación, puede permitir la conexión directa con la plataforma de gestión o necesitar el uso de repetidores o pasarelas. El primer caso lo representan, por un lado, las tecnologías tradicionales (WiFi, 2G, 3G o 4G), las cuales ofrecen una gran cobertura a costa de un alto consumo energético. Por otro lado, tenemos las nuevas soluciones de comunicación específicas para el IoT cuyas redes a día hoy, en la mayoría de las soluciones, no están altamente desplegadas, pero presentan importantes ventajas

como un reducido consumo de energía, diseños baratos, altas coberturas y baja tasa de datos. Como parte de estas soluciones tenemos las denominadas “Low Power Wide Area Network” (LPWAN) tales como Sigfox, LoRa o Weightless entre otras, las cuales han ayudado al éxito inicial del IoT ya que han sido las primeras tecnologías usadas. Estas son poco eficaces en despliegues que requieren elevado tráfico de datos, ya que limitan operación de red al cliente final.

Como caso particular y altamente presente en el IoT hay que tener en cuenta los objetos conectados a dispositivos móviles, funcionando estos como pasarelas hacia la plataforma. Para estas aplicaciones, tecnologías como BLE o NFC están siendo las más utilizadas. A su vez, esta derivación de Bluetooth, BLE, también está siendo utilizada como baliza para el intercambio de información con dispositivos móviles.

2.3.3 Tecnologías aplicativas de red

Puesto que la conectividad de los objetos con las plataformas IoT ha de encaminarse de una forma correcta, existen una serie de tecnologías a nivel aplicativo que permiten realizar esta conexión. En cuanto al encaminamiento, este puede ser a través de protocolos como TCP/IP, a través de tecnologías previas como ZigBee para una posterior traducción a TCP/IP o pueden ir conectadas directamente a la plataforma final. En cuanto a las tecnologías aplicativas de red se dispone, por un lado, de las tradicionales arquitecturas REST sobre servicios web y, por otro lado, de nuevas tecnologías propias del IoT como MQTT sobre TCP, CoAP sobre UDP o 6LowPAN.

2.3.4 Tecnologías Cloud y Big Data

Todos los dispositivos conectados interactúan de una forma u otra con la plataforma IoT o la nube de la aplicación, ofreciendo a éstos datos informativos o de gestión. En este contexto se desarrolla un negocio que inunda el mercado IoT. De cara a los servicios en la nube, empresas como Amazon AWS, Microsoft Azure o IBM Bluemix son las que llevan la iniciativa.

La tendencia de evolución de estas soluciones pasa por integrar distintos niveles de inteligencia a las plataformas IoT, de modo que sean capaces de realizar un análisis de datos y aprendizaje propios del *big data*.

2.3.5 Tecnologías de aplicación IoT

Este grupo de tecnologías se encuentra en el nivel superior de la arquitectura. Estando establecidas las distintas conexiones a los niveles inferiores, en este se ofrece una aplicación, o un conjunto de ellas, con las que el usuario final interactuará.

La función de la aplicación vendrá determinada por el contexto aplicativo en el que nos encontremos. Podemos encontrar aplicaciones de carácter informativo, en las que únicamente se muestra un tratamiento de la información en forma de tablas, gráficas, listas o similares, o

podemos tener una interacción con el sistema a través de esta aplicación. En este último caso, se daría un control al usuario sobre como configurar distintos parámetros de los objetos conectados y diversas funcionalidades a las que se quiera dar permiso.

En definitiva, las tecnologías usadas podrán ser de distintos tipos:

- Aplicaciones nativas de cada uno de los sistemas operativos (Windows, OSX, iOS o Android) sobre ordenadores, tabletas o smartphones.
- Aplicaciones sobre interfaces web compatibles con todas las plataformas.
- Como una combinación de las dos anteriores, existen plataformas IoT y servicios en la nube que dan la posibilidad de crear aplicaciones IoT en sus propios entornos. En este contexto se está desarrollando un nuevo modelo de negocio, como hacen Thingworx o Axeda.

2.4 Aplicaciones

El hecho de dotar de conectividad a un objeto abre un amplio abanico de aplicaciones en el sector IoT. Esta variedad en cuanto a usos se debe principalmente al abaratamiento de hardware de gran capacidad de potencia, a la creciente evolución de los servicios en la nube y al alto despliegue de redes de acceso. Por estos motivos, las aplicaciones tienen enfoques muy variados. Los ámbitos más importantes donde podemos ver aplicaciones de este tipo son:

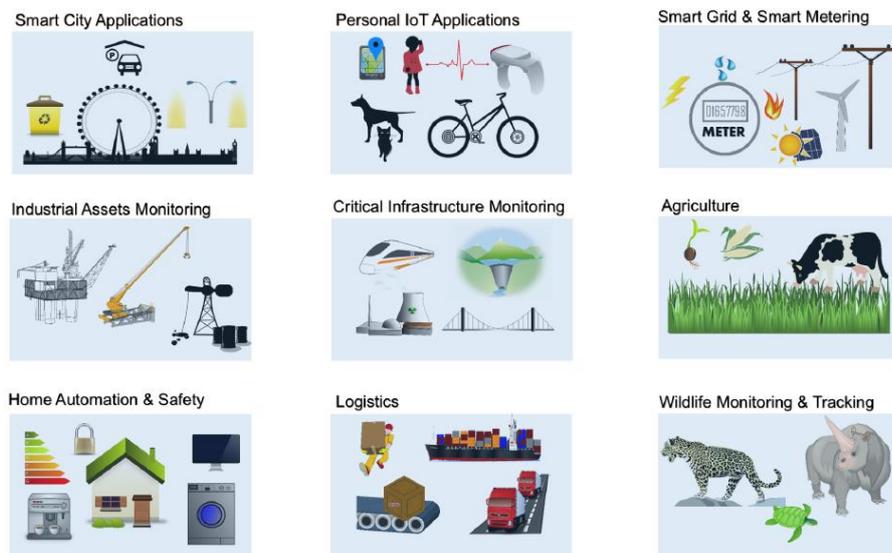


Figura 7 - Ejemplo aplicaciones

2.4.1 Ciudades inteligentes

Una ciudad inteligente se define como aquella ciudad que aplica las tecnologías de la información y de las comunicaciones (TICs) con el objetivo de mejorar la calidad de vida de sus habitantes y asegurar un desarrollo sostenible económico, social y ambiental en mejora permanente. A su vez, este tipo de ciudades se adaptan en tiempo real a las necesidades de los ciudadanos, proporcionándoles información, soluciones y servicios orientados a resolver los problemas generados por la propia urbe.

Este propósito se logra a través de la integración de infraestructuras con sistemas de gestión inteligentes basados en las tecnologías IoT. De este modo, se dota a la ciudad de sensores capaces de obtener información y compartirla en tiempo real.

Como ejemplo de aplicaciones en este ámbito podemos tener desde controles inteligentes del tráfico y la iluminación, la promoción del comercio o la movilidad urbana, hasta la gestión de residuos urbanos.

2.4.2 Domótica

El sector de la domótica tiene un origen anterior al Internet de las Cosas, pero ha visto en este marco un contexto en el que desarrollarse y poder disponer de objetos fácilmente conectables y configurables.

Es tal su relevancia, que las primeras grandes adquisiciones dentro del mercado IoT se produjeron en este ámbito. Por un lado, Google adquirió Nest, una empresa dedicada al desarrollo de termostatos y detectores de humo conectados, y Samsung adquirió Smart Things, la cual propone una solución para la interconexión y el control de todos los dispositivos del hogar.

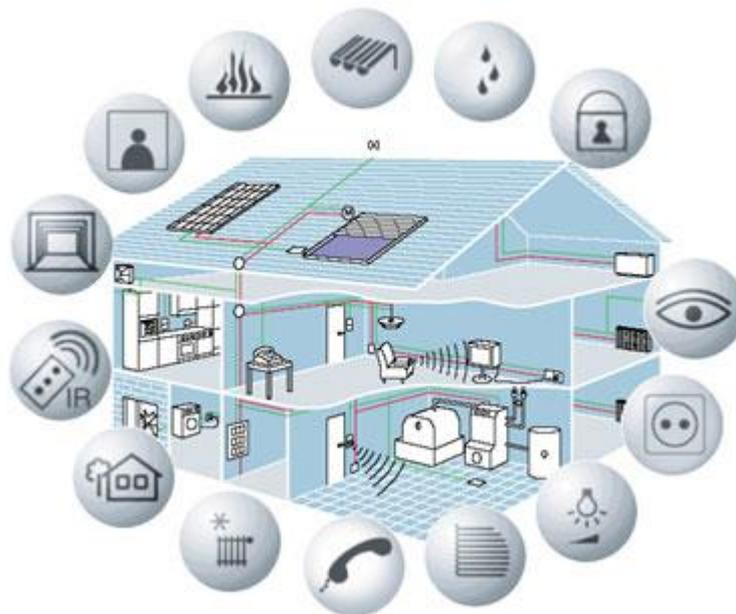


Figura 8 - Domótica

2.4.3 Industria

En el ámbito industrial los sistemas conectados han tenido siempre una gran presencia. La mayoría de ellos trabajan sobre complejos protocolos gestionados por grandes consorcios.

La irrupción del IoT supone una apertura en la conectividad, llevando los datos y la información a la nube y usar el llamado “*big data*” para el análisis y la rápida toma de decisiones.

2.4.4 Salud

Uno de los ámbitos más beneficiados por el desarrollo del sector IoT es el de la salud. Está permitiendo la inclusión y el desarrollo de dispositivos de bajo coste que permiten realizar un seguimiento detallado y específico a los pacientes. Por otro lado, ha permitido, entre otras aplicaciones, la inclusión de sistemas de dosificación de medicamentos y sistemas de ayuda en el hogar y fuera del mismo. De esta manera, el IoT abre un camino en el ámbito de la salud para plantear nuevos usos y aplicaciones sobre el mismo.

2.4.5 Eficiencia energética

Otro sector donde el IoT se está posicionando como fundamental es en la eficiencia energética o en el “*metering*” (medida de consumos) en sectores como el agua, la electricidad o el gas, permitiéndose el análisis continuo en estos campos.

De esta forma, se habilita el desarrollo de dispositivos de medida inteligentes, más conocidos como “*smart meters*”, y de aplicaciones para la monitorización en tiempo real de los mismos. Se permite que los usuarios puedan optimizar su consumo de recursos al tener información detallada en tiempo real.

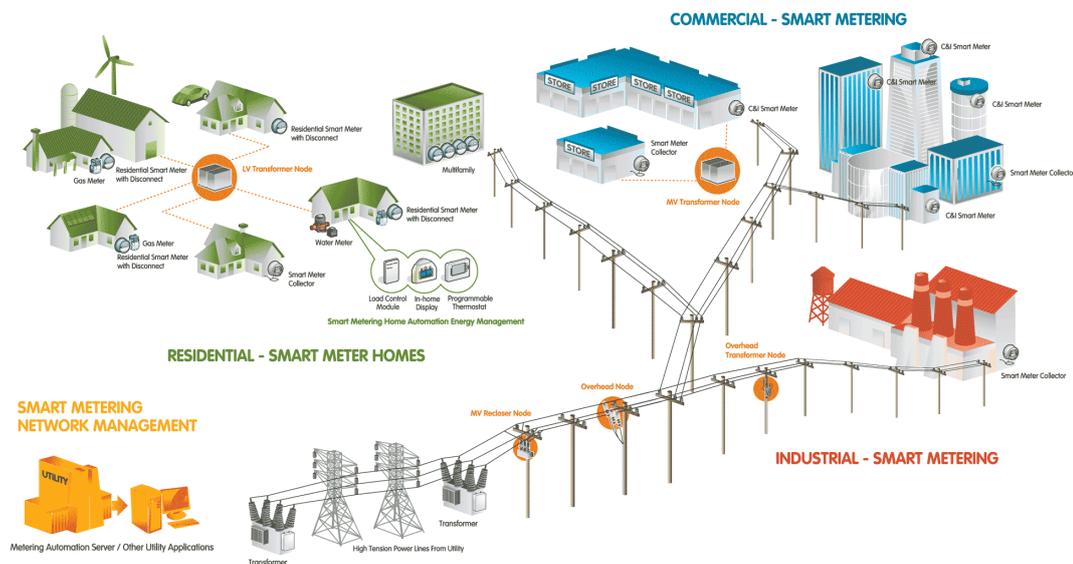


Figura 9 - Medición inteligente

2.4.6 Sector agrícola

El sector agrícola ha sido tradicionalmente uno de los más distantes con las nuevas tecnologías. Pero con la aparición del Internet de las Cosas y las posibilidades de sensorización y actuación remota que este ofrece, se ha convertido en un gran beneficiado. Se están consiguiendo resultados de gran precisión.

La ganadería constituye un primer ejemplo, donde el seguimiento de los animales mediante una combinación de GPS, sensores y conectividad remota permite tener un gran control a distancia sobre el ganado.

Otro ejemplo significativo son las aplicaciones de agricultura de precisión. Gracias a un conjunto de sensores es posible determinar los cuidados óptimos para obtener la mayor producción o la detección precoz de plagas.

2.4.7 Turismo

El auge de las plataformas colaborativas y de opinión son el motor del sector turístico, pero con la llegada del IoT se convierte en un sector potencialmente explotable.

Las aplicaciones en este ámbito pasan por mejorar las experiencias del usuario final. Se pueden desarrollar dispositivos que interactúen con el turista a través del Smartphone o, como valor añadido del IoT, mediante objetos conectados en lugares de alto valor turístico. De este modo, se pueden proporcionar funcionalidades que mejoren el ejercicio turístico.

Capítulo 3. Tecnologías de comunicación IoT

Como se ha comentado en el *punto 2.3*, las tecnologías de comunicación en el Internet de las Cosas han surgido como una necesidad de cubrir las carencias que presentan las tecnologías tradicionales (WiFi, 2G, 3G o 4G) al aplicarse en este sector. De este modo, se han desarrollado alrededor de este contexto las llamadas LPWAN, redes de largo alcance y bajo consumo que ofrecen las mejores capacidades a la hora de implementar una arquitectura IoT. Sus principales características son:

- Bajo consumo eléctrico de los dispositivos, por debajo de los 50 μ Watts. Permite tener objetos conectados durante largos periodos de tiempo con baterías de bajo coste
- Bajo coste de conectividad
- Bajo coste en la fabricación de los dispositivos
- Grandes rangos de cobertura, hasta 60 km por estación base

Las tecnologías dentro del modelo LPWAN presentan pues características excelentes para un amplio rango de aplicaciones de IoT. Como hemos visto en el *punto 1*, este sector se caracteriza por el envío de un reducido número de paquetes de información de pequeño tamaño y por la necesidad de una autonomía prolongada a la hora de sensorizar ciertas zonas, permitiendo así una independencia de la red eléctrica y una despreocupación por la recarga de los dispositivos. Estas necesidades son perfectamente cubiertas y además estas redes LPWAN aportan un largo alcance como valor añadido a las necesidades del IoT, disminuyendo así el número de estaciones que se necesitan para ofrecer cobertura al usuario final.

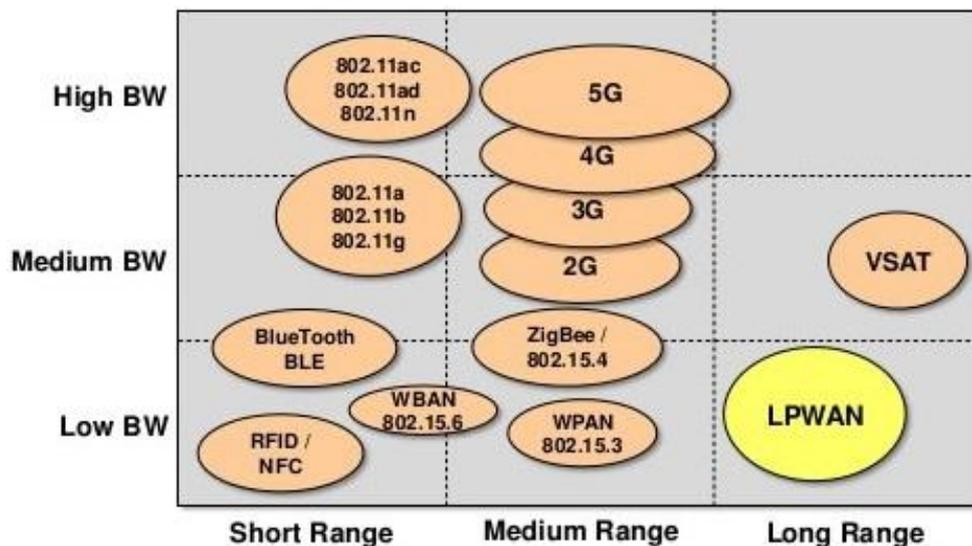


Figura 10 - Situación LPWAN

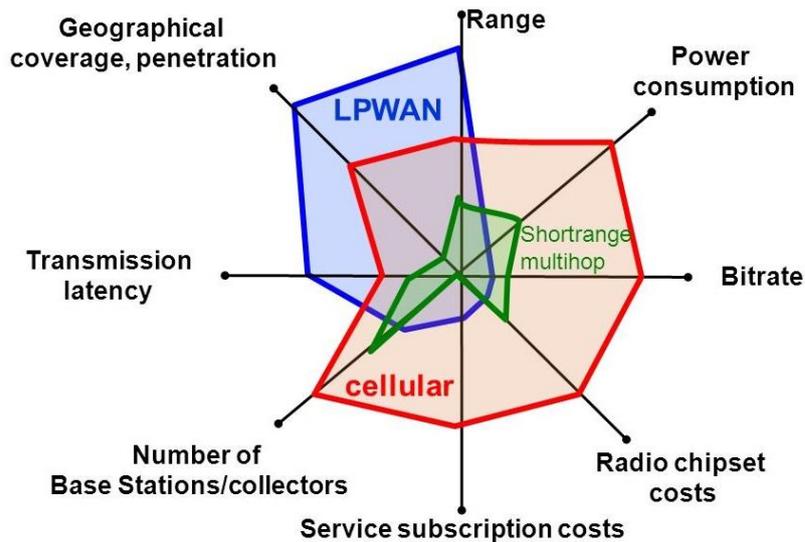


Figura 11 - Comparativa de redes

En la figura 10 y la figura 11 ilustran cómo quedaría situadas las tecnologías LPWAN frente a otras tecnologías de comunicación existentes, enfatizando cómo ofrece características muy interesantes en los puntos clave del sector IoT como son el consumo de energía o la cobertura.

Tras la definición de este modelo de comunicación se han desarrollado una gran variedad de tecnologías que lo siguen, dando lugar a un amplio mercado de negocio. Entre el abanico de posibilidades encontramos desde tecnologías propietarias de empresas, pasando por asociaciones que regulan la suya propia o estándares oficiales.

En este punto veremos algunas de las más utilizadas en la actualidad (RPMA, Sigfox, LoRa, Weightless, DASH7, NB-IoT...) y sus características principales, para finalmente realizar una comparativa de las más importantes y con mayor presencia en el sector. Sus puntos fuertes variarán dependiendo del sector sobre el que centran su negocio.

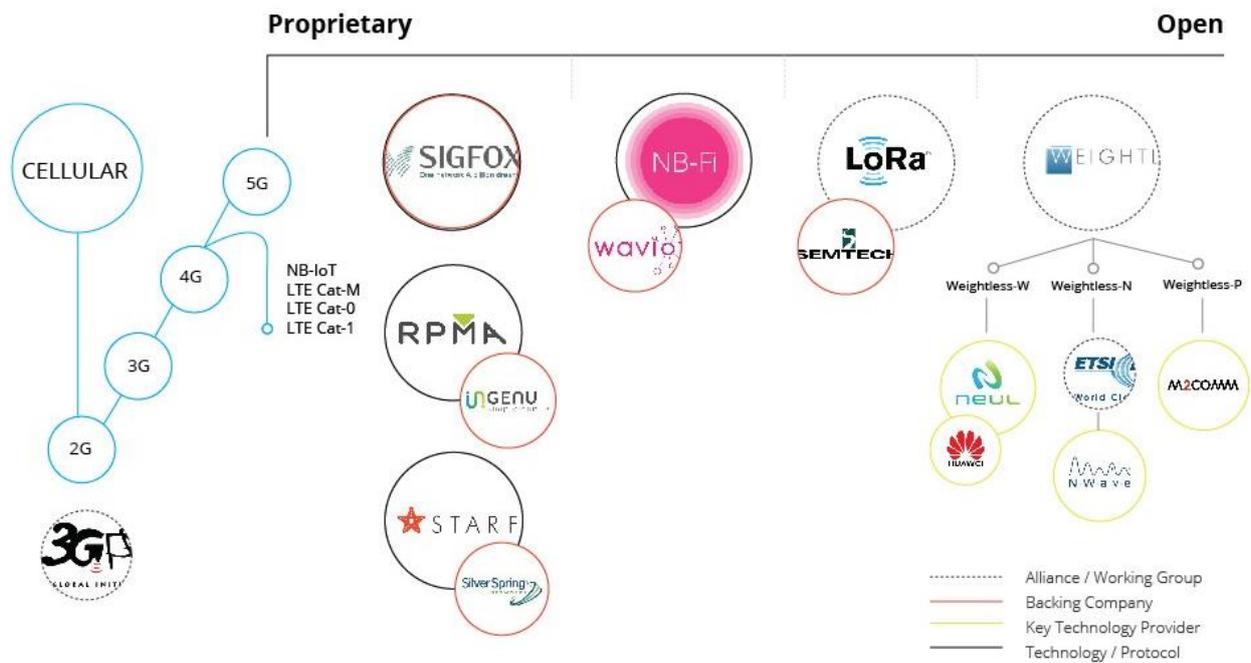


Figura 12 - Situación del mercado IoT

3.1 ETSI LTN



El grupo de estandarización ETSI ha especificado su propia tecnología de comunicación, denominada “*Low Throughput Network*” (LTN), para el sector IoT. Viene definida en tres documentos: Casos de uso para LTN [3], Arquitecturas funcionales [4] y Protocolos e Interfaces [5].

Esta tecnología tiene algunas características que la diferencian de las existentes en el sector. Permite una comunicación bidireccional de largo alcance de hasta 40 km, en entornos libres de obstáculos. Trabaja con un consumo reducido de energía, dando lugar a una elevada vida útil. Y, además, incluye un sistema de procesamiento de la señal de radio para evitar problemas con las interferencias.

Como canal de subida (*Uplink*), trabaja en la banda ISM libre del sub-GHz 868-868,60 MHz, con un ancho de canal de 100 Hz. El esquema de modulación es BPSK su velocidad de modulación es de 100 baudios.

Por otro lado, la norma especifica que el canal de bajada (*Downlink*) usa la banda de 869,40-869,65 MHz. El esquema de modulación utilizado es GFSK con un ancho de canal variable y una velocidad de modulación de 600 baudios.

La potencia de transmisión está limitada. En el caso del enlace de subida, las regulaciones de la región fijan este límite, (típicamente 25 mW), mientras que en la comunicación de bajada, la potencia está limitada a 500 mW

3.2 Sigfox



Una de las tecnologías de comunicación más importantes actualmente en el sector IoT es Sigfox, basada en el estándar ETSI LTN anteriormente descrito. Sigfox es una tecnología propietaria de la empresa que lleva su propio nombre y está basada en una modulación diferencial DBPSK para subida hacia la plataforma y una modulación GFSK para la descarga de datos. En ambas direcciones trabaja sobre la tecnología de comunicación “*Ultra Narrow Band*” (UNB) transmitiendo sobre las bandas de frecuencias de sub-GHz libres.

Los paquetes que transmite son de un tamaño reducido, componiéndose por una parte fija de 12 bytes que incluye un preámbulo, un identificador del dispositivo y otros metadatos, y por la parte variable formada por la información, siendo ésta de hasta 12 bytes. En total, cada paquete puede variar desde los 12 hasta los 24 bytes más unos bits extras usados para la autenticación de los parámetros. A lo largo de un día, Sigfox permite el envío de hasta 140 mensajes, tardando cada uno de ellos aproximadamente 2 segundos en llegar a la estación base.

Una característica a tener en cuenta es que Sigfox es eficaz para las comunicaciones desde los puntos finales a las estaciones base, con una velocidad de 100 bps, pero no es particularmente eficaz en transmisiones desde las estaciones base hasta los puntos finales. Las descargas son más lentas que la subida de información.

El alcance de la tecnología varía según el entorno de aplicación, desde los 30-50 km en entornos rurales hasta los 3-10 km en entornos urbanos.

3.3 LoRa



LoRa es una modulación de radio patentada y desarrollada por Cycleo y adquirida por Semtech en 2012. Actualmente está regulada por LoRa Alliance, formada por diversas empresas con presencia en el sector IoT, que se encarga de promover el uso de esta tecnología. Su arquitectura de comunicación está basada en una conexión a través de pasarelas (*gateways*), las cuales sirven como puerta hacia la plataforma IoT.

Esta tecnología utiliza una estrategia de espectro ensanchado para transmitir a una variedad de frecuencias y velocidades de datos. Esto permite a la pasarela adaptarse a las condiciones cambiantes y optimizar la forma en que intercambia datos con cada dispositivo. Puede transmitir en frecuencias pertenecientes a las bandas de 109 MHz, 433 MHz, 866 MHz y 915 MHz, que ofrecen una menor cantidad de interferencias que otras superiores, como puede ser la banda de 2,4 GHz. En estas frecuencias las señales penetran obstáculos y alcanzan largas distancias con un consumo reducido de energía.

Como consideración importante, LoRa ofrece una funcionalidad bidireccional efectiva, por lo que es buena tanto enviando mensajes desde los dispositivos a las estaciones base como de estas a los puntos finales. Todo ello con una tasa de transferencia entre 300 y 5000 bits por segundo.

3.4 RPMA



“Random Phase Multiple Access” (RPMA) es una solución patentada por Ingenu y diseñada específicamente para las comunicaciones inalámbricas en los contextos M2M e IoT. Principalmente opera en Estados Unidos y basa su tecnología, como bien dice su nombre, en un acceso múltiple de fase aleatoria.

RPMA opera en la banda ISM a 2,4 GHz, lo que hace más sensible a pérdidas e interferencias, hecho que la empresa que gestiona la tecnología supera instalando dos antenas en sus dispositivos. Puesto que la longitud de onda de la banda 2,4 GHz es más menor, es posible diseñar antenas de menor tamaño. Como consecuencia del uso de esta banda de frecuencia, el alcance obtenido es menor y por lo tanto la cobertura se reduce en comparación con las tecnologías que usan las bandas por debajo del gigahercio. Esto se puede traducir en una menor cantidad de dispositivos por estación base.

Al igual que LoRa, RPMA es capaz de soportar una transmisión bidireccional de forma efectiva, con unas velocidades de hasta cientos de miles de bits por segundo, a costa de un mayor consumo de energía que sus competidores.

En cuanto al tamaño de los paquetes que puede enviar, estos pueden variar desde los 6 bytes hasta los 10 kbytes.

3.5 Weightless



Weightless es una de las tecnologías de comunicación más usadas en el ámbito IoT y que da nombre al propio grupo que la regula, Weightless SIG. Este grupo es una organización de estándares sin ánimo de lucro formada para coordinar esta tecnología en el mundo IoT, definiéndola como una tecnología LPWAN de estándar abierto que ofrece gran cobertura, larga duración de las baterías y un reducido coste de fabricación de dispositivos.

Dentro del propio grupo, ofrecen tres variantes de la tecnología: Weightless-N, Weightless-P y Weightless-W, en orden de menos a más complejas.

| | Weightless-N | Weightless-P | Weightless-W |
|-----------------------|---------------------|---------------------|---------------------|
| <i>Directionality</i> | 1-way | 2-way | 2-way |
| <i>Feature set</i> | Simple | Full | Extensive |
| <i>Range</i> | 5km+ | 2km+ | 5km+ |
| <i>Battery life</i> | 10 years | 3-8 years | 3-5 years |
| <i>Terminal cost</i> | Very low | Low | Low-medium |
| <i>Network cost</i> | Very low | Medium | Medium |

Figura 13 - Comparativa Weightless

3.5.1 Weightless-N

Weightless-N, basada en el estándar ETSI LTN, opera en las bandas libres ISM del sub-GHz (169/433/470/780/868/915/923 MHz) usando la tecnología UNB con una modulación digital DBPSK para transmitir dentro de bandas de frecuencia estrechas utilizando un algoritmo de salto de frecuencia para mitigar las interferencias. Proporciona cifrado y autenticación para codificar la información transmitida. La tecnología admite movilidad entre redes, las cuales enrutan automáticamente los mensajes hasta el destino correcto.

Esta tecnología ofrece un rango de alcance de varios kilómetros incluso en ambientes urbanos, un consumo de energía muy bajo y unos costes de hardware y red mínimos. Es por ello que Weightless-N es definida por el propio grupo como un estándar para ser aplicado cuando el bajo coste es la prioridad y cuando una dirección de comunicación es suficiente para nuestra aplicación.

3.5.2 Weightless-P

Weightless-P trabaja en las bandas ISM libres del sub-GHz con una modulación GMSK usando técnicas de transmisión FDMA+TDMA en canales de banda estrecha de 12,5 kHz, siendo su sincronización a nivel de estación base, permitiendo así realizar una programación del uso de los canales de radio. Implementa una reutilización de frecuencias que permite una asignación de canales flexible, mejorando la capacidad de la red y ofrece velocidades de transmisión bidireccionales, tanto de subida como de descarga, entre 200 bps y 100 kbps.

El rango de alcance viene limitado principalmente por la frecuencia a la que trabaja la tecnología de comunicación, siendo estas similares en la mayoría de las soluciones LPWAN. Es por ello que los factores que determinan en última instancia este alcance son la trayectoria de la señal, el presupuesto de la instalación, el tamaño de la antena, la posición y la localización de esta, la tarifa de datos y la energía de transmisión. De este modo, Weightless-P ofrece rangos típicos de 2km en entornos urbanos con un consumo por debajo de los 100 μ W a una potencia de transmisión de apenas 17dBm, permitiendo funcionamientos con una simple pila de reloj.

3.5.3 Weightless-W

Weightless-W es una tecnología de comunicación de estándar abierto que opera en el espectro libre entre las bandas de televisión (400 – 800 MHz), haciendo uso de bandas estrechas. Usa modulación 16-QAM o BPSK con 16 canales FDMA intercalados para América y 24 canales FDMA paralelos para el resto del mundo, a una tasa de 128kbps.

La cobertura de la tecnología es de, como mínimo, 5km en entornos tanto urbanos como abiertos a una velocidad de transmisión entre 1 kbps y 10 Mbps. Dependiendo del presupuesto de la instalación y la aplicación, la vida útil de las baterías puede verse limitada entre los 3 y 5 años. Estas arquitecturas tienen un coste de red e infraestructura mayor que el resto de tecnologías LPWAN a cambio de un aumento de prestaciones.

Por otro lado, los paquetes no están limitados a un tamaño máximo y tienen un tamaño mínimo de 10bytes. A su vez se utiliza encriptación de 128 bits basada en un intercambio de claves secretas.

3.6 NB-Fi



NB-Fi es una tecnología de comunicación patentada por Waviot similar a Sigfox. Opera en las bandas ISM libres por debajo del gigahercio con una modulación DBPSK a una tasa de transmisión entre 10bps y 100bps. Utiliza canales estrechos de 100 Hz de ancho de banda, similar al UNB de Sigfox.

La cobertura que aporta esta tecnología es superior a las de sus competidores del mercado LPWAN. Pese a utilizar la misma modulación y bandas de frecuencias similares para la comunicación, consiguen tener más alcance al usar dispositivos con una mayor potencia de transmisión. De esta forma, su alcance máximo queda fijado en torno a los 16 km para entornos urbanos con un consumo de 166 dBm. En entornos rurales pueden alcanzar hasta los 50 km.

3.7 Grupo 3GPPP



El grupo “3rd Generation Partnership Project” (3GPP) es un acuerdo de colaboración para el desarrollo y evolución de especificaciones del estándar WCDMA.

Este grupo ha estandarizado varias tecnologías de comunicación, agrupándolas en versiones, como puede ser 3G. Dentro del contexto del IoT, 3GPP ha diseñado en los últimos años varios modelos de comunicación:

- **LTE Cat 1:** es la versión 8 del grupo, publicada en 2008. Fue la primera en implantar la tecnología LTE y aunque no fue inicialmente diseñada para el contexto del IoT, sus prestaciones pueden permitir el despliegue de aplicaciones.
- **LTE Cat 0:** es la versión 12 del grupo y fue publicada en 2015. Aporta mejoras frente a la primera versión de LTE, pero sin ser aún específica para el mundo del Internet de las Cosas.
- **LTE Cat M1 (eMTC):** es una de las estandarizaciones de la versión 13 y fue publicada en 2016 pensando en aplicaciones hacia el mundo IoT y M2M.
- **LTE Cat NB1 (NB-IoT):** fue publicada en 2016 y conforma una de las estandarizaciones de la versión 13. Al igual que eMTC, su desarrollo fue focalizado hacia el IoT.

Todas ellas basan su funcionamiento en las actuales redes celulares, aprovechando sus estaciones base y la infraestructura ya desplegada. Como su nombre indica, utilizan la tecnología LTE.

En la siguiente gráfica podemos ver una comparativa de las cuatro nombradas anteriormente, siendo NB-IoT la más adecuada para aplicaciones en el contexto que nos ocupa en el documento.

| | LTE Cat 1 | LTE Cat 0 | LTE Cat M1 (eMTC) | LTE Cat NB1 (NB-IoT) |
|---------------------------------|---------------|---------------------|----------------------|--|
| 3GPP Release | Release 8 | Release 12 | Release 13 | Release 13 |
| Downlink Peak Rate | 10 Mbps | 1 Mbps | 1 Mbps | 250 kbps |
| Uplink Peak Rate | 5 Mbps | 1 Mbps | 1 Mbps | 250 kbps (multi-tone) 20 kbps (single-tone) |
| Number of Antennas | 2 | 1 | 1 | 1 |
| Duplex Mode | Full Duplex | Full or Half Duplex | Full or Half Duplex | Half Duplex |
| Device Receive Bandwidth | 1.08 - 18 MHz | 1.08 - 18 MHz | 1.08 MHz | 180 kHz |
| Receiver Chains | 2 (MIMO) | 1 (SISO) | 1 (SISO) | 1 (SISO) |
| Device Transmit Power | 23 dBm | 23 dBm | 20 / 23 dBm | 20 / 23 dBm |

Figura 14 - Comparativa 3GPP

3.8 DASH7



La “DASH7 Alliance” es un consorcio que define la tecnología de comunicación “DASH7 Alliance Protocol” (D7AP) dentro del ámbito de las LPWAN, proporcionando conectividad de alcance medio para actuadores y sensores de baja

potencia.

D7AP utiliza un esquema de modulación de banda estrecha utilizando GFSK en las bandas ISM libres por debajo del gigahercio. En comparación con el resto de tecnologías de comunicación, DASH7 obtiene una menor latencia debido a su ciclo de funcionamiento, por el cual comprueba el canal periódicamente. Pese a su mejora en latencia, presenta un aumento en el consumo de energía, cuestión de vital importancia en el ámbito IoT.

Con sus características, D7AP permite un alcance de hasta 5 km en entornos urbanos con unas tasas de transferencia de decenas de kbps.

3.9 IEEE 802.11ah



IEEE 802.11ah, más conocido como “WiFi-HaLow”, ha sido desarrollado por el conocido grupo (IEEE) como una especificación para aplicaciones del Internet de las Cosas. Su proceso de normalización comenzó en

noviembre de 2010 y se publicó en 2016.

Esta tecnología opera en las bandas de sub-GHz exentas de licencia, en contraposición a las frecuencias de 2,4 y 5 GHz a las que trabajan el resto de especificaciones de la norma 802.11, mediante la técnica de transmisión OFDM (*Multiplexación por División de Frecuencias Ortogonales*).

De este modo, la velocidad de transmisión de datos va desde los 150 kbps hasta los 346 Mbps, llegándose a alcanzar distancias en torno a 1 km.

3.10 Comparación

Una comparativa a nivel general de las tecnologías más importantes la podemos encontrar en la *tabla 1*. Los valores han sido obtenidos a partir de la información suministrada por los propietarios de dichas tecnologías en distintos medios.

Como se puede ver, las características de cada una de ellas son muy variadas. Los enfoques varían desde tecnologías que se centran en obtener un mayor rango de cobertura a costa de perder enteros en cuanto a consumo energético, hasta tecnologías que invierten esta situación, teniendo una elevada vida útil de las baterías pero un corto alcance. Entre ellas también existen casos en los que se mantiene un equilibrio de prestaciones sin arriesgar en ningún punto de ellos. El tercer factor a considerar es la tasa de bits, que varía desde los 100 bps hasta los 350 Mbps.

| Name of standard | Weightless | | | SigFox | LoRa | LTE-Cat M | IEEE 802.11ah | DASH7 | RPMA | NB-Fi |
|---------------------------------|-------------------------------|---------------------------|---------------------|-----------------------------------|--|------------------------|--|--------------------------------------|--------------------------------------|----------------------------------|
| | -W | -N | -P | | | | | | | |
| Frequency Band | TV whitespace (400 – 800 MHz) | Sub-GHz ISM | Sub-GHz ISM | 868/902 MHz ISM | 433/868/780/915 MHz ISM | Cellular | License-exempt bands below 1GHz, excluding TV White Spaces | 433, 868, 915 MHz ISM/SRD | 2.4 GHz ISM | 868 MHz |
| Channel Width | 5 MHz | Ultra narrow band (200Hz) | 12.5 kHz | Ultra narrow band (100 Hz) | EU: 8x125 kHz, US: 64x125 kHz/8x125 kHz | 1.4 MHz | 1/214/8/16 MHz | 25 kHz or 200 kHz | 1MHz | Ultra narrow band (100 Hz) |
| Range | 5 km (urban) | 3 km (urban) | 2 km (urban) | 30-50 km (rural) 10 km (urban) | 2-5 km (urban) 15 km (rural) | 2.5-5 km | Up to 1km (outdoor) | 0-5 km | 4.6 km (urban) | 16,6 km (urban) 50 km (rural) |
| End Node Transmit Power | 17 dBm | 17 dBm | 17 dBm | 10 μ W to 100 mW | EU: <+ 14 dBm, US: <+ 27 dBm | 100 mW | 1 mW to 1W | Depending on FCC/ETSI regulations | to 20 dBm | 14 dBm |
| Packet Size | 10 byte min | Up to 20 bytes | 10 byte min | 12 bytes | Defined by user | 100-1000 bytes typical | with Aggregation: Up to 65,535 Bytes, w/o: Up to 7,991 Bytes | 256 bytes max/packet | 6-10 kbytes | Defined by user |
| Uplink Data Rate | 1kbps to 10 Mbps | 100 bps | 200 bps to 100 kbps | 100 bps to 140 messages/day | EU: 300 bps to 50 kbps US: 900-100 kbps | 20 kbps | 150 kbps - 346.666 Mbps | 9.6 kbps, 55.55 kbps or 166.667 kbps | AP aggregates to 624 kbps per sector | 10-100 bps to 32 meesages/day |
| Downlink Data Rate | 1kbps to 10 Mbps | No downlink | 200 bps to 100 kbps | Max 4 messages of 8 bytes/day | EU: 300 bps to 50 kbps US: 900-100 kbps | 20 kbps | 150 kbps - 346.666 Mbps | 9.6 kbps, 55.55 kbps or 166.667 kbps | AP aggregates to 156 kbps per sector | 10-100 bps to 32 meesages/day |
| Devices per Access Point | Unlimited | Unlimited | Unlimited | 1M | Uplink: > 1M Downlink: < 100 k | 20 k | 8191 | NA (connectionless communication) | Up to 384,000 per sector | > 1M |
| Topology | Star | Star | Star | Star | Star on Star | Star | Node-to-Node, Star, Tree | Star, Tree | Star, Tree | Star |
| End Node Roaming Allowed | Yes | Yes | Yes | Yes | Yes | Yes | Allowed by other IEEE 802,11 amendments | Yes | Yes | Yes |
| Governing Body | Weightless SIG | | | Sigfox | LoRa Alliance | 3GPP | IEEE 802.11ah working group | DASH7 Alliance | Ingenu | Waviot |

Tabla 1 - Comparación LPWANs

Uno de los conceptos más importantes a la hora de elegir la tecnología más adecuada es el alcance. En la figura 15 podemos ver una comparativa entre las más utilizadas y el rango de cobertura que pueden llegar a ofrecer en un entorno urbano. El alcance viene determinado por la frecuencia de portadora de la señal, el tipo de modulación y la potencia de transmisión. En cuanto a esta última, es evidente que cuanto más “fuerza” tenga la señal, la cobertura será mayor. Pero no es un concepto que deba estudiarse solo. En la comparación de la figura 15 podemos ver como la tecnología que utiliza Ingenu, RPMA, y la que utiliza Waviot, NB-Fi, no son muy dispares en cuanto a potencia, tienen una diferencia de apenas 3 dBm, pero el alcance es radicalmente superior cuando se usa la tecnología NB-Fi. Esto se debe, como podemos ver en la figura 16, a la frecuencia de portadora que utiliza la tecnología. A frecuencias de 2,4 GHz, como es el caso de RPMA u otras tecnologías como WiFi, el nivel de penetración sobre objetos fijos como paredes, es muy reducido. Por el contrario, a frecuencias de 868 MHz la propagación de la señal no sufre tanta pérdida de potencia, llegando a posibilitar que tecnologías como la utilizada por Waviot lleguen a los 16 km de alcance en entornos urbanizados.

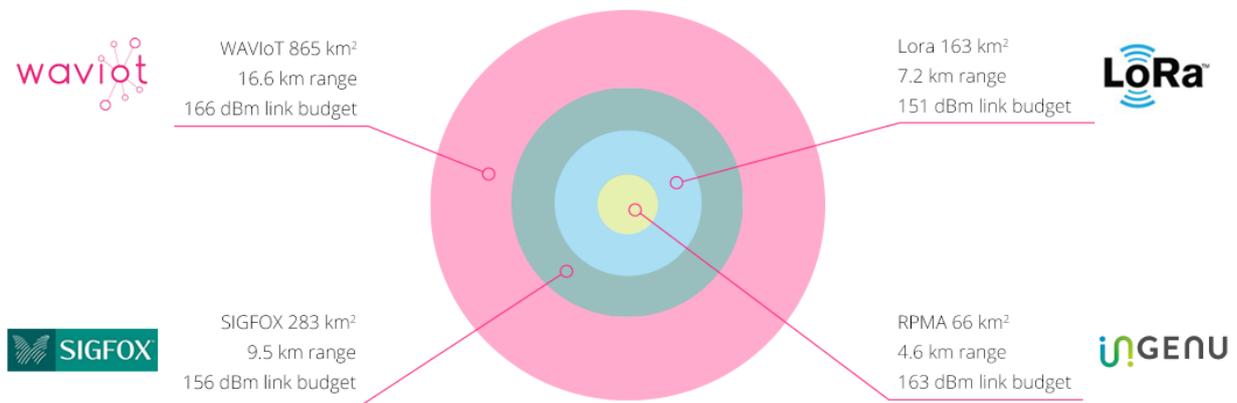


Figura 15 - Comparación alcances

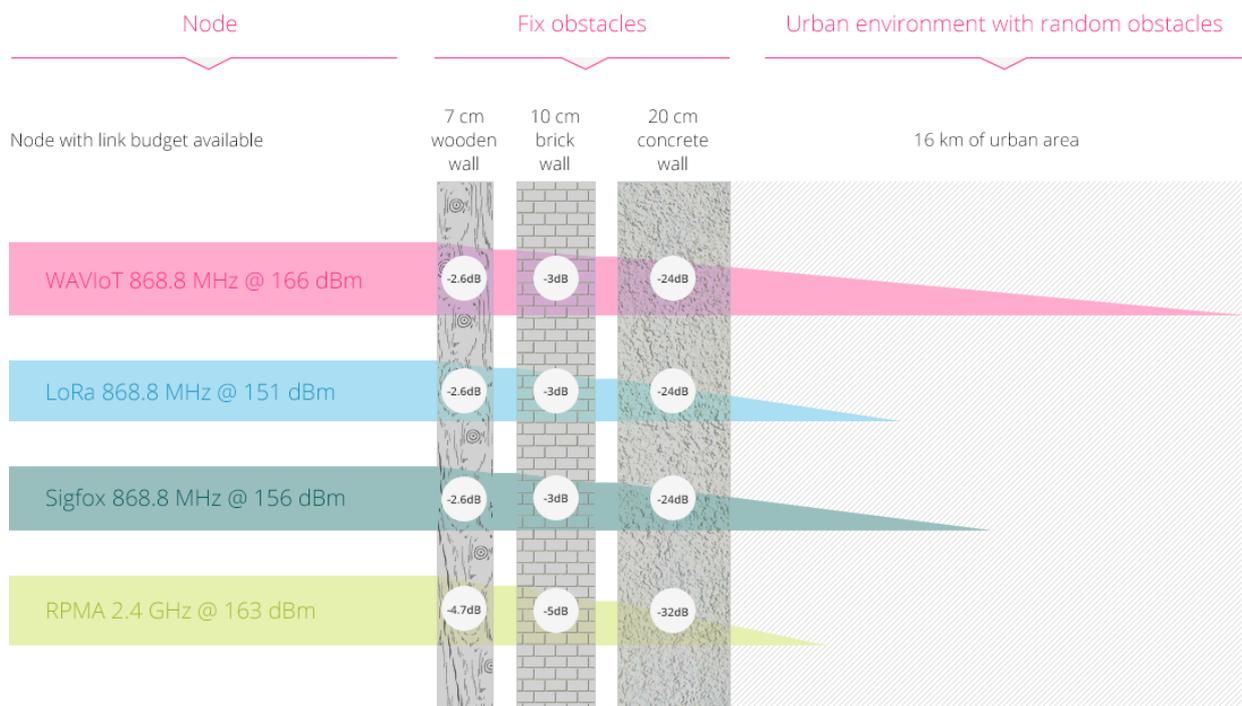


Figura 16 - Comparativa penetración

En la misma línea, es preciso señalar que las potencias de transmisión usadas en las comparativas de las ilustraciones anteriores no son fácilmente alcanzables y suponen un consumo de energía muy elevado para una aplicación realista del sector IoT. Un dispositivo empotrado que integre una de las tecnologías de comunicación anteriormente descritas apenas llegará a los 30 dBm de potencia. De este modo, si se realiza la comparativa con estos niveles de potencia, asumibles por dispositivos comerciales, los valores de alcance de las distintas tecnologías se verán drásticamente reducidos.

3.11 Caso real

Un ejemplo de esta relación entre potencia y alcance, y su considerable diferencia con las prestaciones ofrecidas por los fabricantes, ha sido estudiada dentro de este presente trabajo como un caso real de comunicación mediante estas tecnologías.

Se han utilizado dos módulos LoRa Dragino [16] sobre dos placas Arduino Due para estudiar el alcance de esta tecnología, cuya frecuencia de comunicación se sitúa en los 868 MHz. La potencia de transmisión que podía ofrecer nuestro sistema empotrado estaba en el rango de 10 a 23 dBm, siendo establecida a su potencia máxima.

Las pruebas se han realizado bajo el modelo TCP y UDP. Puesto que integrar este protocolo de comunicación requería implementar un nuevo módulo en el arduino, se ha optado por programar código Arduino que funcionara de forma similar. De este modo se han creado cuatro programas: un transmisor y un receptor para TCP y otra pareja de ellos para UDP.

El código del transmisor TCP envía paquetes de reducido tamaño al receptor. Este, si recibe el paquete, contestará con un ACK a nivel de aplicación. Si el reconocimiento llega al transmisor, seguirá el mismo proceso con un segundo paquete, así hasta 100 de ellos. En el caso de que no se reciba contestación, ya sea porque el reconocimiento se ha perdido o no ha llegado el paquete al receptor, se realizan cinco reintentos de envío bajo un timeout de espera de 500 ms. Si tras esta serie de reenvíos sigue sin obtenerse información, se trata el paquete como perdido y se continúa con el ciclo de la transmisión. En este caso, el transmisor medirá el tiempo que ha tardado en enviar todos los paquetes y recibir sus correspondientes ACKs, y el receptor medirá el nivel de señal con el que recibe los paquetes (RSSI) y el número de ellos que ha conseguido obtener.

En cuanto al código UDP, es una implementación sencilla del anterior. El transmisor únicamente envía 100 mensajes de pequeño tamaño y calcula el tiempo que ha tardado en enviarlos. El receptor, anotará el número de paquetes que ha recibido y sus correspondientes RSSIs.

Con las placas configuradas, se han realizado pruebas a distintas distancias para analizar la cobertura de esta tecnología. Como dato a tener en cuenta en ellas, la vivienda a la que se va a hacer referencia es un sexto piso situado a una altura de aproximadamente 22 metros con respecto a la calle. Estas pruebas son y están numeradas de la siguiente forma:

- **Prueba 1:** transmisor y receptor juntos a menos de un metro.
- **Pruebas 2 y 3:** transmisor y receptor situados en habitaciones colindantes
- **Pruebas 4 y 5:** una placa situada en el exterior y otra en el interior, a la misma altura.
- **Pruebas 6 y 7:** una placa en el interior de la vivienda y otra en el garaje de la finca.
- **Prueba 8:** prueba adicional para UDP con una placa en el exterior de la casa y la otra a la vuelta de la esquina.
- Distintas pruebas UDP con una placa en el exterior de la vivienda y la segunda a distintos kilómetros de distancia.

Las últimas pruebas, en las que se comenzaba a transmitir a varios kilómetros de distancia y nos íbamos acercando hacia el nodo receptor, únicamente se realizaron bajo el modelo UDP, ya que tenían como propósito establecer el alcance de comunicación de la tecnología. Por ello, en la *tabla 3* aparece la prueba adicional número 8, en la que se materializa el primer momento de recepción de información de este estudio de distancia.

Los resultados obtenidos, tanto para TCP como UDP pueden verse en las tablas siguientes:

TCP

| Número de prueba | Distancia (m) | Paquetes totales | % Paquetes perdidos | Nº reenvíos | Tiempo envío (s) | RSSI medio (dBm) |
|------------------|---------------|------------------|---------------------|-------------|------------------|------------------|
| 1 | 0,5 | 100 | 0 | 0 | 21,6 | -30,17 |
| 2 | 2 | 100 | 0 | 0 | 21,6 | -45,74 |
| 3 | 2 | 100 | 0 | 0 | 21,816 | -50,44 |
| 4 | 15 | 100 | 0 | 3 | 23,586 | -73,76 |
| 5 | 15 | 100 | 0 | 1 | 22,262 | -75,79 |
| 6 | 22 (vertical) | 100 | 0 | 1 | 22,262 | -92,81 |
| 7 | 22 (vertical) | 100 | 0 | 4 | 24,464 | -92,45 |

Tabla 2 - Prestaciones LoRa (TCP)

UDP

| Número de prueba | Distancia (m) | Paquetes totales | % Paquetes perdidos | RSSI medio (dBm) |
|------------------|---------------|------------------|---------------------|------------------|
| 1 | 0,5 | 100 | 0 | -27,72 |
| 2 | 2 | 100 | 0 | -45,61 |
| 3 | 2 | 100 | 1 | -44,09 |
| 4 | 15 | 100 | 0 | -75,94 |
| 5 | 15 | 100 | 1 | -80,6 |
| 6 | 22 (vertical) | 100 | 1 | -94,63 |
| 7 | 22 (vertical) | 100 | 3 | -92,56 |
| 8 | 30 (diagonal) | 100 | 22 | -90,61 |

Tabla 3 - Prestaciones LoRa (UDP)

Los resultados obtenidos denotan una clara pérdida de prestaciones con el aumento de la distancia entre los nodos, como era de esperar. Esta tendencia a la baja está claramente representada con el RSSI, el cual indica la potencia con la que llega la señal al receptor, estando el límite de una comunicación fiable alrededor de los -90 dBm.

Como se ha comentado inicialmente el alcance de la tecnología LoRa dista mucho de las especificaciones dadas debido a la diferencia de potencia. El alcance según los reguladores es de 7,2 km a 151 dBm de potencia. En nuestra prueba, apenas se ha conseguido llegar a las varias decenas de metros con un dispositivo comercial, como es Arduino.

De este modo, aplicando una comunicación en un entorno urbano y real, se demuestra que el alcance que podemos obtener con esta tecnología dista mucho de los valores aportados por sus propietarios. Y de igual forma, esta situación es extrapolable al resto de tecnologías de comunicación.

Capítulo 4. ETSI - Low Throughput Network (LTN)

Como se ha visto en el *punto 3.2*, Sigfox está basado en el estándar de comunicación desarrollado por la ETSI para tecnologías de comunicación LPWAN: *Low Throughput Network*. Es por ello que para el objetivo del presente documento es necesario conocer en profundidad el funcionamiento y los detalles de dicho estándar para realizar una correcta implementación.

Como ya se ha comentado en el *punto 3.1*, el estándar de comunicación ETSI LTN viene definido en tres documentos de especificaciones: El documento [3] especifica los casos de uso del estándar, [4] detalla la arquitectura funcional y el documento [5] especifica concretamente el funcionamiento del protocolo y sus interfaces.

4.1 Arquitectura

La arquitectura de una red LTN viene definida en la *figura 17*. En ella se define la arquitectura global y las diferentes interfaces que la componen.

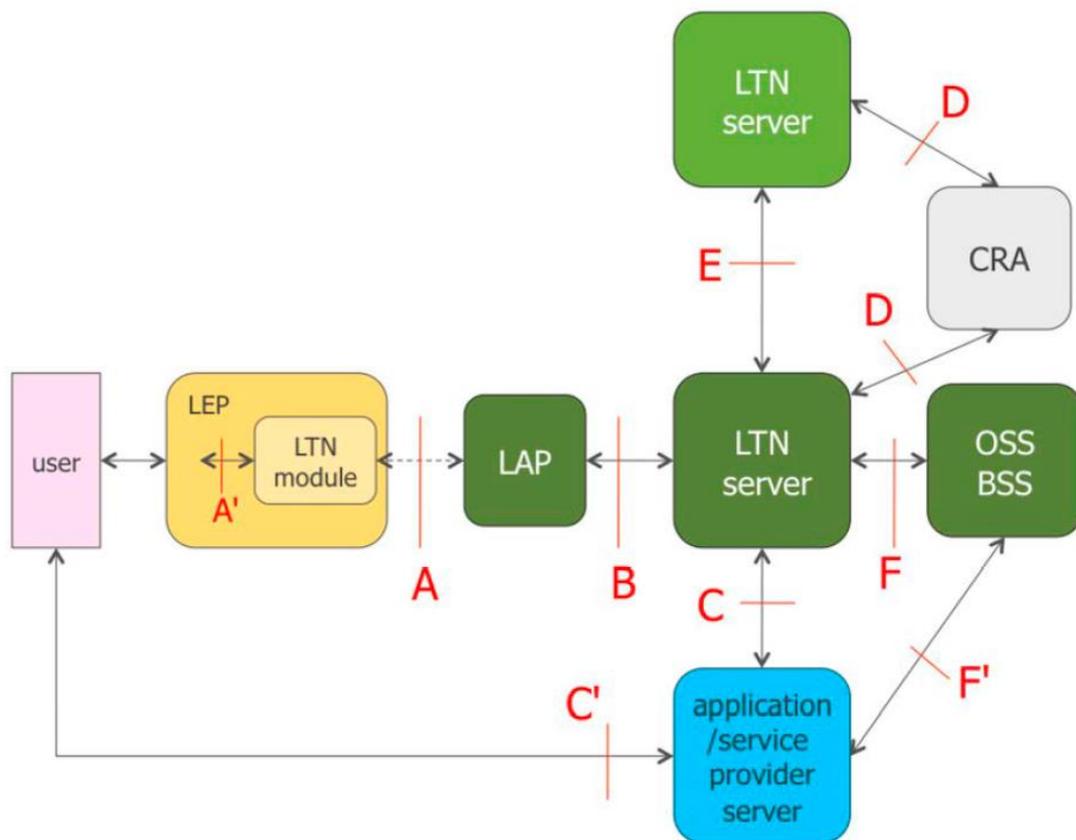


Figura 17 - Arquitectura LTN

Los distintos puntos que componen la arquitectura de una red LTN están formados por:

- **Interfaces:** representan las conexiones entre los distintos elementos de la red
- **LEP:** es el punto final de la comunicación; la conexión con el usuario
- **LAP:** es el punto de acceso de la red LTN
- **Servidores LTN**
- **Proveedores de servicios**
- **OSS:** sistema de soporte de operaciones
- **BSS:** sistema de soporte de las empresas
- **CRA:** autoridad central de autorizaciones

Siguiendo la definición de la arquitectura de red de la *figura 17*, todos los componentes de ésta están comunicados entre ellos por distintos interfaces. Cada uno de estos tiene sus propiedades específicas.

Los interfaces B, C, D, E, F y F' están definidos bajo el estándar de red de área amplia ("Wide Area Network, WAN"), fuera del ámbito de este trabajo. La función de estas interfaces son las siguientes:

- **Interfaz B:** esta interfaz ofrece conexión entre los distintos LAPs de la red y los servidores LTN. Este enlace puede estar establecido en uno de los distintos tipos de comunicación propios de redes WAN, como pueden ser GPRS o ADSL
- **Interfaz C:** la interfaz C comunica los servidores LTN con los distintos proveedores de servicios. Esta comunicación se basa en los principios REST y sus APIs
- **Interfaz D:** establece el enlace entre los servidores LTN y el CRA, de modo que permite la personalización de los puntos finales de la red. Se puede atribuir a los LEPs identificadores, claves secretas y códigos de autorización
- **Interfaz E:** es un enlace directo entre distintos servidores LTN de diferentes operadores. Un ejemplo de uso puede ser para intercambiar datos entre operadores en "roaming"
- **Interfaz F:** esta interfaz comunica los sistemas OSS/BSS con los servidores LTN. Se utiliza para activar los registros o recoger los datos de carga de los LEPs, o para recuperar el estado de la red.
- **Interfaz F':** la interfaz F' comunica los sistemas OSS/BSS con los proveedores de servicios, con el objetivo de permitir gestionar los registros y los datos de los LEPs

El interfaz C' es un enlace de comunicación multicanal entre el usuario y el proveedor de servicios, cuyo estudio queda fuera del presente proyecto.

Por otro lado, A' es una interfaz interna de un LEP entre el módulo LTN y la aplicación del usuario, cuya definición también queda fuera de este documento.

Finalmente, la interfaz A representa la comunicación de radiofrecuencia entre los distintos LEPs y LAPs. Puesto que uno de los principales objetivos del presente proyecto es estudiar el alcance de una red LTN entre un punto de acceso y un dispositivo final, este interfaz va a ser definido en el *punto 4.2*.

4.2 Interfaz A

Dentro de la interfaz A hay dos tipos de implementaciones que utilizan el mismo espectro, pero con diferentes aproximaciones:

- Banda ultra estrecha (“*Ultra Narrow Band*”, UNB)
- Espectro ensanchado de secuencia ortogonal (“*Orthogonal Sequence Spread Spectrum*”, OSSS)

Ambas técnicas de comunicación están representadas en la *figura 18* y están localizadas en el espectro según las regulaciones de cada región. En España, esta regulación viene especificada en el boletín oficial del estado [21], por la que se determina la banda de frecuencias 868 a 870 MHz. En Europa la banda de comunicación es la misma.

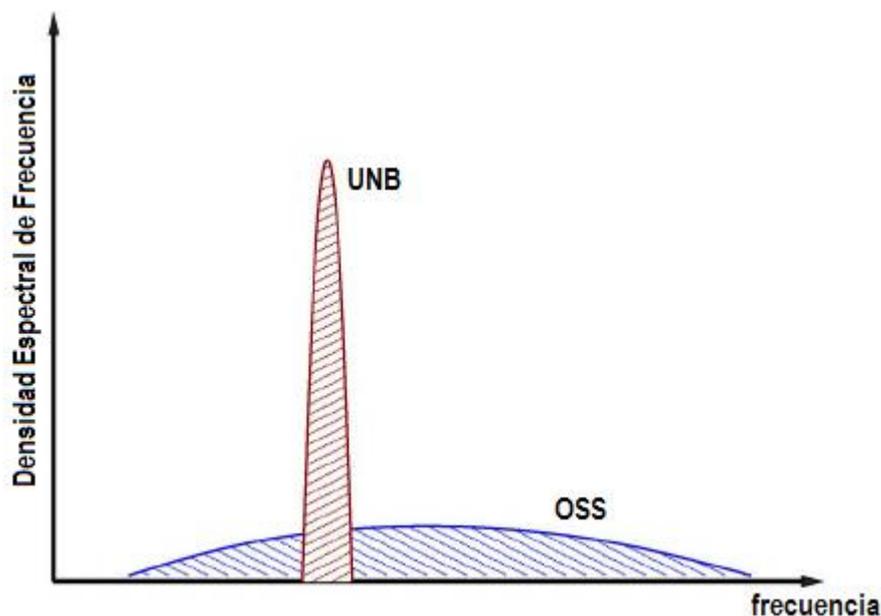


Figura 18 - UNB vs OSSS

4.2.1 UNB

Algunos estándares de comunicación del sector IoT, como Sigfox o LTN, utilizan la tecnología de transmisión UNB. Este modo de comunicación, como se puede observar en la *figura 18*, concentra la potencia de transmisión en una banda estrecha del espectro, típicamente de cientos de hertzios, obteniendo así un alto nivel de densidad espectral de potencia con un nivel de potencia reducido, lo que permite alcanzar distancias de hasta decenas de kilómetros.

El coste de este gran alcance es un ancho de banda muy reducido, permitiendo únicamente la transmisión de pequeñas cantidades de información. De este modo, no es posible enviar un gran volumen de datos en una única comunicación. Por tanto, UNB está indicado únicamente para aplicaciones que requieran la transmisión de pequeñas cantidades de información.

ETSI LTN implementa una comunicación bidireccional basada en UNB con diferencias entre los enlaces de subida (comunicación de los LEPs en dirección a los LAPs) y de bajada (comunicación desde la red en dirección a los LEPs) con respecto a las tramas y su sincronización.

Además, implementa RFTDMA como método de acceso al medio, similar a ALOHA y muy diferente a los convencionales métodos de contención o predicción de colisiones.

4.2.1.1 Método de acceso al medio (RFTDMA)

RFTDMA aprovecha el reducido ancho de banda de la comunicación para establecer una gran cantidad de canales a lo largo de la banda de espectro asignada, cuyo valor es típicamente de unos cientos de kilohertzios. De este modo, en cada transmisión se escoge un canal aleatorio y se realiza el envío de información sin preocuparse por la ocupación de dicho canal.

De esta forma, por ejemplo, para un ancho del espectro de 12 KHz y un ancho de canal de 100 Hz tendríamos un total de 120 canales disponibles, de los cuales en cada transmisión se selecciona uno aleatoriamente. Esta aleatoriedad, sumada al reparto de las transmisiones en el tiempo y a la corta duración de la ocupación de los canales, hace muy difícil que se produzca una colisión. En la *figura 19* podemos ver un ejemplo de la ocupación de los distintos canales en el tiempo durante la transmisión de varios nodos. [22]

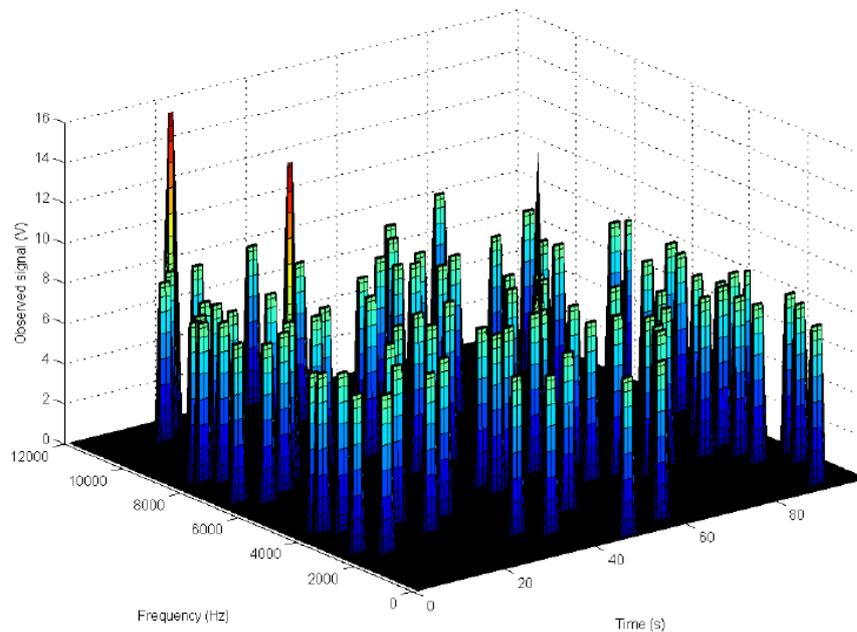


Figura 19 - Ejemplo RFTDMA

4.2.1.2 Enlace de subida UNB

Las especificaciones del enlace de subida son las siguientes:

- **Máscara del canal:** 100 Hz (600 Hz en EEUU)
- **Tasa de transmisión:** 100 baudios (600 baudios en EEUU)
- **Modulación:** BPSK
- **Potencia de transmisión:** de acuerdo a las regulaciones de cada región. Típicamente un máximo de 25 mW
- **Sensibilidad del receptor:** -135 dBm de sensibilidad mínima recomendada
- **Banda de frecuencia:** 868,00 – 868,60 MHz

El formato de la trama MAC de subida es el representado en la *figura 20*:

- **Preámbulo:** indica que es una trama UNB de subida. Tamaño de 4 bytes
- **Campo de sincronización:** permite la sincronización de los dispositivos. Tamaño de 2 bytes
- **Identificador:** identifica el punto final (LEP). Tamaño de 4 bytes
- **Carga de datos:** contiene la información a transmitir. Tamaño variable de 0 a 12 bytes
- **Autenticación:** permite autenticar los dispositivos. Longitud variable
- **FCS:** permite verificar la integridad de la información. Longitud de 2 bytes (CRC)

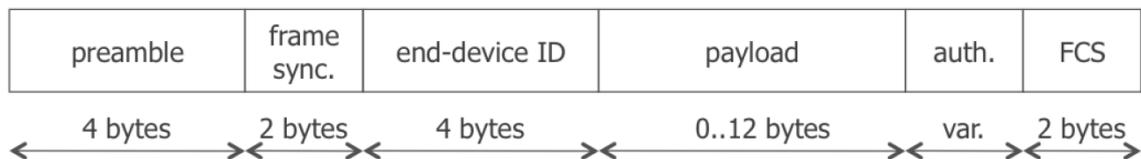


Figura 20 - Trama UNB MAC de subida

4.2.1.3 Enlace de bajada UNB

Las especificaciones del enlace de bajada son las siguientes:

- **Máscara del canal:** selección dinámica
- **Tasa de transmisión:** 600 baudios
- **Modulación:** GFSK
- **Potencia de transmisión:** 500 mW
- **Banda de frecuencia:** 869,40 – 869,65 MHz

El formato de la trama MAC de bajada es el representado en la *figura 21*:

- **Preámbulo:** indica que es una trama UNB de subida. Tamaño de 32 bits
- **Campo de sincronización:** permite la sincronización de los dispositivos. Tamaño de 13 bits
- **Bandera:** tamaño de 2 bits
- **FCS:** permite verificar la integridad de la información. Longitud de 8 bits (CRC)
- **Autenticación:** permite autenticar los dispositivos. Tamaño de 16 bits
- **Código de error:** permite indicar si ha ocurrido un error y el tipo de este. Longitud variable
- **Carga de datos:** contiene la información a transmitir. Tamaño variable

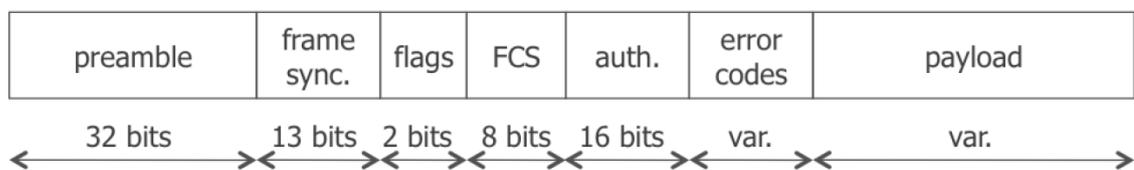


Figura 21 - Trama UNB MAC de bajada

4.2.1.4 Sincronización entre enlaces de subida y bajada

Después de cada transmisión de subida por parte de un LEP, éste abre una ventana de recepción de duración fija en la que habilita la recepción de mensajes. La red LTN transmitirá los mensajes de bajada dentro de esta ventana de comunicación, teniendo que ser la duración de esta definida previamente en la configuración de la red.

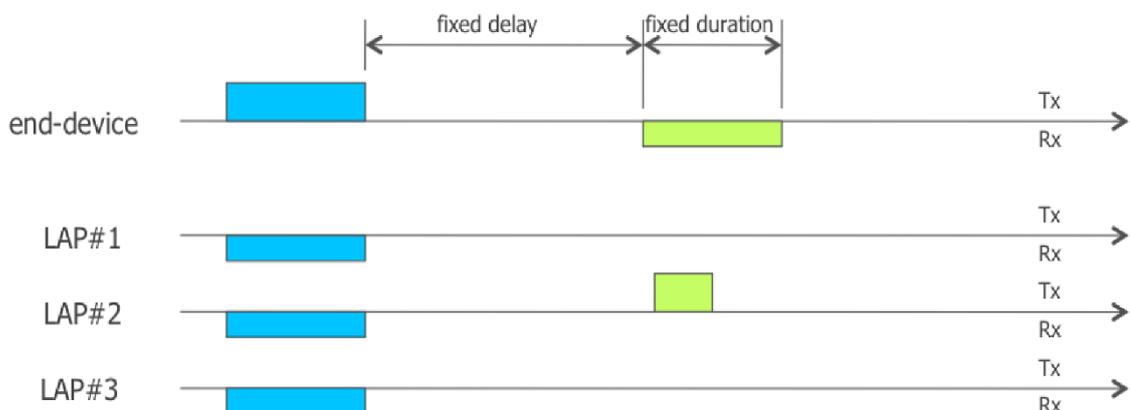


Figura 22 - Sincronización UNB

4.2.2 OSSS

Algunas tecnologías de comunicación, como por ejemplo LoRa, utilizan un modelo de espectro ensanchado. ETSI LTN define un esquema de transmisión denominado OSSS, donde la potencia de transmisión está repartida a lo largo del espectro tal como se puede observar en la *figura 18*. Con este esquema se obtiene una densidad espectral de potencia menor, obteniéndose un alcance más reducido en comparación a UNB, pero con un mayor ancho de banda, permitiendo en el envío de mayores cantidades de información con respecto al modelo de banda estrecha.

De esta forma, una interfaz de comunicación del tipo A implementada mediante OSSS se convierten en un modelo de transmisión efectivo en caso de una gran densidad de nodos, ya que proporciona una alta eficiencia espectral, incluso en casos de potencia de transmisión limitada.

Al igual que para el modelo UNB, LTN establece una comunicación bidireccional para sistemas basados en OSSS. En este caso, tanto las características de los enlaces como los formatos de la trama de las capas física y MAC son únicos para ambas direcciones.

4.2.2.1 Especificaciones del radio enlace

Como se ha comentado anteriormente, las especificaciones de los enlaces de subida y bajada son comunes para ambas direcciones de comunicación, y son las siguientes:

- **Máscara del canal:** depende del ensanchamiento del espectro. Desde 8 kHz hasta 500 KHz
- **Tasa de chips:** 8 kcps – 500 kcps. Representa el número de bits por segundo de la señal
- **Tasa de datos:** 30-50000 bps
- **Modulación:** equivalente a DSSS con señalización ortogonal
- **Sensibilidad:** mínima sensibilidad recomendada de -135 dBm
- **Potencia de transmisión:** limitada por las regulaciones regionales
- **Banda de frecuencia:** 868 – 870 MHz

4.2.2.2 Formato de trama física

El formato de trama de la capa física está representado en la *figura 23*:

- **Preámbulo:** tamaño de 12 símbolos (4 para sincronización y 8 para detección de la señal)
- **Cabecera:** incluye el tamaño del paquete, una bandera para indicar una posible fragmentación, un campo para indicar el tipo de codificación y una última bandera para indicar si se usa CRC
- **CRC de cabecera:** integra un CRC de la cabecera para comprobar la integridad. Junto a la cabecera suman un tamaño de 20 bits
- **Carga de datos:** contiene la trama MAC. Tamaño variable

| Field Size | 12 symbols | 20 bits | | variable |
|------------|------------|-----------------|------------|------------------|
| Field Name | Preamble | Physical Header | Header CRC | Physical Payload |

Figura 23 - Trama OSSS física

4.2.2.3 Formato de trama MAC

La trama MAC del modelo OSSS sigue las recomendaciones de IEEE 802.15.4, cumpliendo esta trama con el formato definido en dicho estándar. La figura 24 establece los campos de la trama MAC:

- **Control de trama:** tamaño de 2 bytes. Formado por los campos:
 - *Tipo de trama:* bits del 0 al 2
 - *Habilitación de seguridad:* bit 3
 - *AR:* bit 4
 - *Habilitación de ADR:* bit 5
 - *Modo de direccionamiento destino:* bits 6-7
 - *Modo de direccionamiento fuente:* bits 8-9
 - *Indicador de más datos:* bit 10
 - *Tipo de protocolo de transporte:* bits 11-12
 - *Habilitador de extensión del campo de control de trama:* bit 13
 - *Versión de la trama:* bits 14-15
- **Contador de trama:** tamaño de 0 o 2 bytes
- **Dirección destino:** tamaño de 0, 2 u 8 bytes
- **Dirección fuente:** tamaño de 0,2 u 8 bytes
- **Extensión del campo de control de la trama:** tamaño de 0 o 2 bytes
- **Carga de datos:** tamaño variable
- **Encriptación HASH del mensaje:** 4 bytes

| Field Size (in Bytes) | 2 | 0/2 | 0/2/8 | 0/2/8 | 0/2 | variable | 4 |
|-----------------------|---------------|---------------|---------------------|----------------|-------------------------|---------------|-------------|
| Field Name | Frame Control | Frame Counter | Destination Address | Source Address | Frame Control Extension | Frame Payload | Hash Crypto |

| Bit Position | 0-2 | 3 | 4 | 5 | 6-7 | 8-9 | 10 | 11-12 | 13 | 14-15 |
|--------------|------------|------------------|----|-------------|-----------------------|------------------------|-----------|----------------------|---------------------------------|---------------|
| Field Name | Frame Type | Security Enabled | AR | ADR Enabled | Dest. Addressing Mode | Source Addressing Mode | More Data | Transported Protocol | Frame Control Extension Enabled | Frame Version |

Figura 24 - Trama OSSS MAC

4.2.2.4 Sincronización entre enlaces de subida y bajada

Dentro de una red con el sistema de comunicación OSSS existen dos tipos de nodos (LEPs) definidos según su capacidad para recibir información, es decir, su forma de sincronización con los LAPs:

- Los nodos del tipo A abren una o varias ventanas de recepción después de cada transmisión. La duración de cada una de estas ranuras temporales está predefinida por la red.
- El segundo tipo de nodos, los B, mantienen una sincronización derivada de los “*beacons*” enviados por los LAPs cada 128 segundos. De este modo, estos nodos escuchan cada cierta ranura de tiempo predeterminada esperando un mensaje.

Por lo tanto, la sincronización en redes con el modelo OSSS puede realizarse de dos formas diferentes según el tipo de nodos involucrados en la red. En redes con UNB únicamente existe un tipo de sincronización igual a realizada por los nodos A.

Capítulo 5. Simuladores de redes de telecomunicación

En ocasiones contar con un sistema real para experimentar sobre él puede llegar a tener un elevado coste o directamente ser imposible de implementar.

La evaluación del comportamiento de las redes de transmisión de datos puede realizarse mediante diversos métodos, siendo la simulación uno de los más empleados por su bajo coste, su adaptabilidad y su capacidad para someter al sistema bajo estudio a las condiciones más diversas.

Actualmente en el sector de la investigación, y más concretamente en investigaciones centradas en redes y protocolos de comunicación, la elección de un buen simulador cobra una vital importancia.

Por lo tanto, se tiene que para realizar estudios específicos para evaluar el desempeño de diferentes sistemas se utilizan simuladores que permitan diseñar experimentos. Estos deben garantizar homogeneidad en las respuestas y una alta confiabilidad en los resultados y las conclusiones que arroje un determinado estudio.

La diferencia entre los diferentes simuladores que podemos encontrar para el análisis de redes de comunicación, y en general para todo tipo de sistemas, radica en el método que utilizan para realizar la simulación. Los dos tipos más comunes son:

- **Simuladores de eventos discretos:** funcionan modelando el sistema como una secuencia de eventos de manera que cada uno de ellos sucede en un instante de tiempo determinado. Incluye la elección de un parámetro de incremento o intervalo de tiempo básico sobre el cual se desarrollará el proceso. Estos simuladores son óptimos para la simulación de sistemas secuenciales o que utilicen colas.
- **Simuladores de tiempo continuo:** funcionan utilizando ecuaciones diferenciales y modelos matemáticos que describen la evolución del sistema de forma continua. Su complejidad es elevada ya que necesitan definir los elementos y las interacciones entre ellos con un gran nivel de detalle, siendo las influencias de unos sobre otros inmediatas. Estos simuladores son de gran utilidad cuando se desea analizar un proceso que cambia muy sutilmente con el tiempo, como puede ser un sistema movido por agua.

De este modo, sabiendo los tipos de simuladores, en nuestro presente trabajo, en el cual simulamos redes de telecomunicación, utilizaremos un simulador de redes de eventos discretos. Los más utilizados, y que mayores prestaciones ofrecen al usuario, son los siguientes.

5.1 NCTUns

El simulador “*National Chiao Tung University, Network Simulator*” (NCTUns) es un simulador de sistemas de telecomunicación sobre Linux. Simula y emula redes entrando en el Kernel del sistema operativo. Su desarrollo se basa en el simulador NS y está preparado tanto para la simulación de arquitecturas sencillas como complejas, como pueden ser redes GPRS, ópticas o satelitales.

Es un simulador que se caracteriza por utilizar una sintaxis sencilla a la hora de describir los parámetros, la topología y la configuración de la simulación, ya que implementa una interfaz gráfica para realizarlo. De este modo, hace el entorno muy intuitivo para su manejo y acerca al usuario a la implementación de los modelos de red que se simulan.

NCTUns se utiliza principalmente para la simulación de redes móviles o inalámbricas, permitiendo el estudio de los niveles de calidad de servicios (QoS) de las señales del sistema. También puede emular dispositivos reales, tales como hubs, puntos de accesos o switches.

Además, también permite el diseño, la evaluación y la implementación de distintos tipos de protocolos y aplicaciones como IEEE 802.3, IEEE 802.11, IP, UDP/TCP o RTP/RTCP.

A la hora de programar, como ya se ha comentado, se utiliza una interfaz de usuario gráfica que, además de cubrir esta función, nos permite visualizar el desarrollo de la simulación mediante animaciones. A su vez, ofrece la posibilidad de representar los resultados gráficamente y estadísticamente, incluso permite la consulta de datos en tiempo de ejecución sin necesidad de cancelar la simulación.

5.2 Kiva

Kiva es un simulador de redes basado en java que está orientado a la simulación de redes bajo el protocolo IP, ARP e ICMP. Ofrece una API que permite utilizar las funciones de simulación desde otras aplicaciones Java.

Se centra en el estudio de los datagramas y de su encaminamiento y tratamiento. También permite simular el funcionamiento de Ethernet a nivel de capa física.

Este simulador presenta una interfaz gráfica que permite diseñar la red, desde la topología hasta el direccionamiento a través de la modificación de las tablas de encaminamiento. Es por ello que se centra en el estudio de los datagramas y su encaminamiento y tratamiento, simulando además pérdidas de paquetes o fallos en el enrutamiento.

De este modo, Kiva se compone de dos partes:

- **API:** ofrece las funciones de simulación de redes a otras aplicaciones.
- **Interfaz gráfica:** permite la configuración de la red (topología, direccionamiento, encaminamiento de la información, etc)

Es por ello, que este simulador es uno de los más completos a la hora de simular redes de comunicación basadas en IP.

5.3 COMNET III

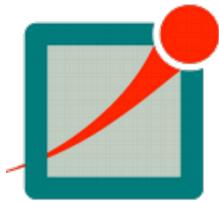
COMNET III es un simulador comercial desarrollado por CACI Products Inc. cuyo lenguaje de programación está basado en MODSIM II.

Presenta una interfaz gráfica de usuario que utiliza un método de ventanas con menús y barras de herramientas. Permite describir el modelo de red a través de esta interfaz asociando los distintos elementos del sistema de acuerdo a las características del modelo a implementar.

A la hora de simular, ofrece la posibilidad de ver el modelo de red y el intercambio de mensajes entre los nodos de la red de manera dinámica en tiempo de ejecución. Además, ofrece resultados como la ocupación del enlace, colisiones o número de mensajes generados.

Estas características lo convierten en un simulador orientado al aprendizaje a bajo nivel, ya que no ofrece una amplia flexibilidad a la hora de diseñar un modelo.

5.4 OMNeT++



OMNeT++ es un simulador de eventos discretos, modular y orientado a objetos. Su utilización está enfocada al modelado de tráfico en redes de telecomunicaciones y al modelado de protocolos, así como a la evaluación de rendimiento de sistemas de software.

Está basado en C++ y se dedica principalmente a la creación de simulaciones de redes cableadas e inalámbricas orientadas en diferentes ámbitos como: redes de sensores, redes Ad-Hoc o redes fotónicas, entre muchas otras.

El simulador tiene soporte para Windows y en varias distribuciones de Unix soportando diversos compiladores de C++ y es uno de los más utilizados en la comunidad investigadora y académica.

OMNeT++ permite una descripción detallada de las características del sistema a simular, desde aplicaciones de red o protocolos de enrutamiento, hasta infraestructuras de red. Esto le da la capacidad de proporcionar diferentes modelos de movilidad, contemplando diferentes escenarios, con la posibilidad de cambiar su comportamiento en función de la aplicación.

5.5 OPNET MODELER



OPNET MODELER es un simulador de redes comercial creado por OPNET ("Optimized Network Engineering Tools") basado en eventos y orientado a objetos, donde cada uno de estos tiene una lista de atributos definidos.

Puede ejecutarse en sistemas operativos Windows y Unix.

Actualmente es uno de los simuladores más avanzados, ya que su funcionamiento se caracteriza por una evolución temporal de la simulación cuyo comportamiento está ligado a instantes concretos en los que los sistemas realmente intervienen.

OPNET MODELER ofrece una gran flexibilidad a la hora de realizar modificaciones sobre el código fuente de las librerías, permitiendo diseñar las propias para nuestros modelos de simulación. A su vez, permite dotar de movilidad a los nodos de la red, definir tipos de tráfico y obtener parámetros sobre los paquetes, los enlaces o la potencia de las señales.

A la hora de realizar el modelado, el simulador ofrece una interfaz gráfica basada en 3 editores organizados de una manera jerárquica:

- **Editor de proyectos:** representa gráficamente la topología de la red y permite configurar los protocolos.
- **Editor de nodos:** describe el flujo de datos entre los elementos de la red.
- **Editor de procesos:** soporta las especificaciones, protocolos y parámetros a modelar de la red.

Estas características lo convierten en un simulador muy completo, permitiendo simular aplicaciones y protocolos como: HTTP, TCP, IP, BGP, RIP, RSVP, Ethernet, UMTS o ATM, entre otros.

5.6 NS-2



“*Network Simulator 2*” (NS-2) es un simulador de eventos discretos pensado para el desarrollo de redes a un gran nivel de detalle. Se basa en el lenguaje de programación C++ y en OTcl para implementar scripts para la configuración, y es compatible con plataformas Unix y Windows. Es de código abierto y es ampliamente

utilizado en investigación.

La interfaz de usuario está basada en un editor por código, mediante el cual se diseñan y se configuran las simulaciones. A su vez, puede incorporar otras aplicaciones que le aportan nuevas funcionalidades tales como la representación gráfica de los resultados de las simulaciones o la obtención de valores estadísticos de las mismas.

NS-2 implementa todo tipo de redes de comunicación y permite la simulación de una gran variedad de protocolos de cualquier capa del modelo OSI como: IEEE 802.3, IEEE 802.11, RTP, TCP, UDP, HTTP, TELNET o CSMA/CA.

Por todas sus características, sumado a que es una de las herramientas de simulación más antiguas, se convirtió en uno de los simuladores más ampliamente usados, posición que ha perdido en favor de su sucesor, NS-3.

5.7 NS-3



“*Network Simulator 3*” (NS-3) es un simulador basado en eventos discretos, desarrollado en C++ como sucesor de NS-2. Presenta un fundamento idéntico al de su predecesor, pero es importante saber que NS-3 y NS-2 no son extensiones compatibles entre ellas.

Al igual que NS-2, posee la capacidad de simular todo tipo de redes, desde cableadas hasta inalámbricas, permitiendo un profundo modelado de todos los elementos que conforman la red del modelo a simular.

5.8 Comparativa

A la hora de elegir el software para el desarrollo de nuestro trabajo, se han ido descartando opciones mediante una comparación de las capacidades de cada uno. Por un lado, NCTUns, Kiva o COMNET III ofrecen una flexibilidad muy reducida, pudiéndose emular muy pocos aspectos de una red, como por ejemplo en el caso de Kiva, que únicamente se centra en el estudio de los datagramas y el encaminamiento y tratamiento de estos, lo cual no es suficiente para estudiar una tecnología de comunicación LPWAN.

Por otro lado, OPNET MODELER y NS-2 ofrecen un nivel de flexibilidad muy similar a NS-3 pero presentan detalles que fueron decisivos para descartarlos. El primero de ellos era de pago, por lo que no era de interés comprar una licencia para utilizar un simulador que ofreciera las

mismas características que un software libre. Y en el caso del segundo, debido a la llegada de NS-3, se ha convertido en un simulador obsoleto.

Llegado al punto final, la elección queda entre NS-3 y OMNeT++, los cuales ofrecen características similares, convirtiéndose en los más utilizados en la comunidad académica. Ambos son libres de licencia, basados en C++ y ofrecen un alto rango de posibilidades a la hora de describir todo tipo de redes, así como sus infraestructuras. Los factores que los diferencian radican a un nivel más cercano al usuario. Siendo, como opinión personal, la experiencia de programación de NS-3 más cómoda, debido a su sencillo modo de programación e inclusión de los distintos modelos y al intuitivo control del flujo de los programas de simulación.

Por ello, en el presente documento se ha elegido NS-3 como el simulador para realizar nuestro proyecto debido a su condición de código abierto, a su alta flexibilidad, al alto número de librerías ya implementadas y a su sencillo manejo a la hora de programar.

En el *capítulo 6* será explicado detalladamente.

Capítulo 6. Network Simulator 3 (NS-3)

A la hora de verificar cualquier teoría sobre el funcionamiento de una red, o de probar su comportamiento bajo ciertas condiciones, es conveniente utilizar un simulador. Los simuladores de la familia “*Network Simulator*” (NS-1, NS-2 y NS-3) son una serie de simuladores para redes basados en eventos discretos.

El origen y desarrollo de esta familia comenzó en 1989 con NS-1 como una variante del “*REAL Simulator*” y actualmente sostenido por voluntarios sin ánimo de lucro.

Posteriormente, se desarrolló NS-2, llegando a convertirse en el simulador más utilizado por la comunidad académica e investigadora en el área de redes, ya que implementa una gran cantidad de módulos para dotar de funcionalidades a las redes simuladas. La última versión, 2.34, se presentó al público el 17 de junio de 2009.

La última versión de esta familia, NS-3, fue creada en 2008 y actualmente es el simulador más utilizado e implementa todas las funcionalidades de su predecesor NS-2.

6.1 Características

Como se ha comentado en el *punto 5.7*, NS-3 es un simulador de redes basado en eventos discretos y programado en el lenguaje C++. Su software es totalmente libre, licenciado bajo la licencia GNU GPLv2., y aunque es el sucesor de NS-2, el software de NS-3 fue creado desde cero, buscando una mayor fidelidad con la realidad y un mayor orden en las librerías, de modo que se permitiera una modificación sencilla de éstas.

Por lo tanto, el núcleo de NS-3 está organizado en librerías C++. Cada una de ellas contiene las clases necesarias para simular un tipo de programa, protocolo o modelo en concreto, como puede ser un enlace punto a punto o una red wifi.

Una de sus características más importantes, y la que le hace destacar frente al resto, es el hecho de que está orientado a generar grandes cantidades de datos a partir de una simulación. De este modo se puede observar con un gran detalle el comportamiento de la red o modelo simulado. Para hacer estas simulaciones, es necesario crear un script en C++ o en Python, en el cual se definen todos los objetos de la simulación, los parámetros de éstos, la relación entre ellos y las características generales de la simulación.

6.2 Comparación con NS-2

La principal diferencia entre el simulador NS-2 y el NS-3 se encuentra en el modo de programación y la visualización de los resultados. En NS-2, todo el código se genera en un script en formato OTcl que genera datos que son visualizados mediante herramientas gráficas incluidas dentro del propio simulador, como *Network Animator*. Por el contrario, en NS-3 también se utilizan scripts para realizar la simulación, pero en este caso el simulador genera archivos del tipo *.pcap*, los cuales pueden ser interpretados por varios programas de capturar de paquetes, como puede ser *Wireshark*.

El simulador NS-3 contiene todos los modelos que posee NS-2, y un añadido de nuevos modelos de propagación, modelos inalámbricos y protocolos de comunicación, entre otros. Este hecho denota una mayor capacidad de simulación con respecto a su predecesor.

A su vez, la versión 3 de la familia NS, cubre el conjunto de fallos que presentaba la segunda versión. Entre estos fallos destacan la falta de versatilidad, debido a una dependencia entre modelos, el deficiente uso de técnicas de programación orientadas a objetos y el rígido acoplamiento entre C++ y OTcl.

6.3 Organización del software

El simulador NS-3, como bien se ha dicho ya, se basa en eventos discretos, es decir, en sucesos instantáneos que suceden en un determinado momento temporal y que pueden cambiar el estado del sistema. Su comportamiento se caracteriza por una secuencia, finita o infinita, de estados delimitados por eventos que ocurren de manera asíncrona. Este tipo de sistema puede verse representado en la siguiente figura.

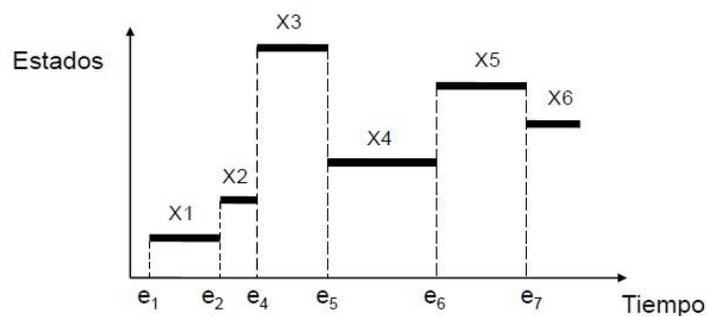


Figura 25 - Sistema de eventos discretos

La estructura del simulador se encuentra dividida en distintos niveles, como puede verse en la figura 26, la cual se encuentra organizada dentro del directorio principal (*/src*).

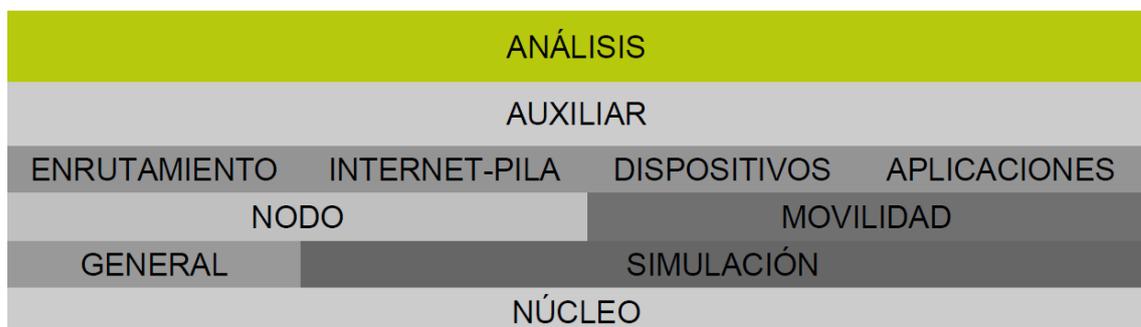


Figura 26 - Organización software NS-3

A la hora de simular y realizar una implementación es necesario, más allá de tener los conocimientos en el lenguaje de programación C++, conocer los contenidos del simulador, y lo que nos ofrece cada uno de ellos, y la ubicación de estos en los distintos directorios.

En primer lugar tenemos en los niveles uno y dos de la organización (*figura 26*) tres módulos de simulación claves. El núcleo de la simulación (*src/core*) que se utiliza como base para crear el motor de simulación (*src/simulator*) y el módulo de generación de paquetes (*src/packet*), los cuales son los objetos fundamentales en un simulador de red. Estos tres módulos conforman la base de toda simulación y son independientes de la red y de los modelos específicos de los dispositivos.

En el tercer nivel se encuentra la base de toda topología de red, los módulos nodo (*src/node*) y movilidad (*src/mobility*). Ambos módulos utilizan modelos específicos para instalar la configuración y almacenar los datos de la misma. Estos parámetros pueden ser desde el tipo de movilidad en el caso del segundo módulo, o el tipo de nodo, ya sea inalámbrico o alámbrico, en el caso del segundo, entre otros parámetros de configuración.

En el cuarto nivel encontramos al conjunto de módulos destinados a incorporar ciertas funcionalidades y aplicaciones a la red, desde el tipo enrutamiento o los efectos de propagación de la señal de radio, hasta la inclusión de protocolos como puede ser IP.

En los dos últimos niveles encontramos los módulos auxiliares y de análisis, los cuales contienen los comandos fundamentales para que el núcleo pueda desarrollar las aplicaciones configuradas en los niveles inferiores.

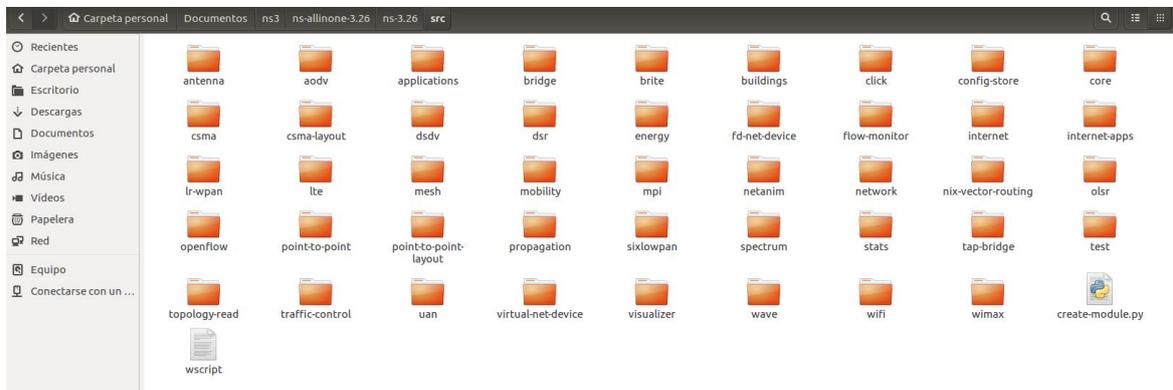


Figura 27 - Directorio /src

6.3.1 Orientación a objetos

Al estar NS-3 programado fundamentalmente en el lenguaje C++, su orientación está basada en objetos.

Estos objetos son elementos básicos que representan todos los componentes incluidos en una simulación. El simulador dispone de las clases *ns3::Object* y *ns3::ObjectBase* para proveer una administración eficiente de la memoria y permitir que los objetos puedan ser agregados entre sí.

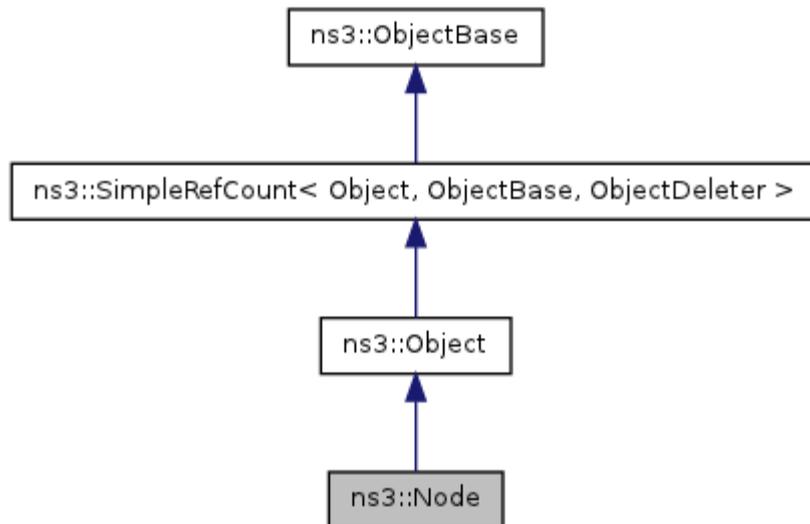


Figura 28 - Modelo de objetos

NS-3 hace uso de punteros inteligentes, los cuales simulan el comportamiento de un puntero normal. Estos colaboran a evitar la pérdida de memoria o a realizar una liberación automática de recursos. Su implementación se realiza mediante la clase `ns3::Ptr< >`. Por ejemplo, el crear un nodo se realiza de la siguiente forma: `Ptr<Node> nodo = CreateObject<Node> ();`

A su vez, los objetos pueden ser agregados entre sí, evitando la construcción de clases de un tamaño elevado y que cumplan con todas las funcionalidades. Por ello, cada clase posee una función específica y limitada. De esta forma, el nodo que se ha creado en el párrafo anterior, sería creado sin ninguna funcionalidad; estaría vacío. Para incluirlo en una aplicación específica habría que agregarle los objetos necesarios para que cumpla con los requisitos de la simulación.

6.3.2 Estructura de un script NS-3

Los scripts son un conjunto de instrucciones almacenadas en un archivo de texto que, para su ejecución, son interpretadas línea a línea en tiempo real. A su vez, pueden estar embebidos en otros lenguajes para aumentar sus funcionalidades.

NS-3, como ya se ha comentado en puntos anteriores, basa su ejecución en scripts característicos del simulador. La estructura de estos es detalla a continuación.

Cabecera del script

La primera línea especifica el tipo de codificación que utilizará el código fuente.

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
```

En las siguientes líneas se especifica la información acerca de la licencia de distribución del software NS-3. Esta información aparece comentada, al igual que la codificación en la línea anterior.

Módulos del script

Antes de proceder a la programación del script, es necesario incluir los módulos que va a utilizar como soporte. Estos incluyen todos los archivos de inclusión pública en sus respectivos módulos.

```
#include "ns3/core-module.h"
#include "ns3/simulator-module.h"
#include "ns3/node-module.h"
#include "ns3/helper-module.h"
```

Definición de nombres en NS-3

Todo script del simulador tiene que iniciarse con la declaración NS-3, la cual permite un manejo más amplio de las variables y de la inclusión de funciones.

```
using namespace ns3;
```

Registros del script

Al script se le añade una sentencia que permite indicar o registrar los procesos que realiza la simulación mediante una serie de mensajes, ya sean de error o de información, entre otros tipos.

```
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
```

Función principal del script

La manera de iniciar un programa NS-3 sigue la línea establecida por su lenguaje de programación, C++. Se declara una función principal (*main*) del programa.

```
int
main (int argc, char *argv[ ])
{
}
```

Como componente añadido, en el instante que se inicia la escritura del programa, se han de añadir las especificaciones sobre los registros para que se pueda dar información de la simulación durante su ejecución. Esto se realiza de la siguiente manera:

```
LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);  
LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
```

Clases

Dentro del estilo de programación característico de un script NS-3 existen las llamadas clases. Estas se utilizan en el simulador para realizar el proceso de declaración de variables, tipos de redes de datos o tipos protocolos de comunicaciones, entre una amplia variedad de funcionalidades.

Algunas de las más importantes son: *NodeContainer*, *PointToPointHelper*, *NetDeviceContainer*, *InternetStackHelper* o *Ipv4AddressHelper*.

Dentro de la variedad de clases, existen un tipo especial de ellas llamadas *Helpers*. Estas, generalmente, se encargan de instalar funcionalidades en un objeto.

Simulador

Una vez diseñado el programa y la aplicación, es necesario iniciar el proceso de simulación. Esto se lleva a cabo con la siguiente línea:

```
Simulator::Run ();
```

Cuando el sistema lea esta instrucción, ejecutará todas las tareas que se han programado durante el script siguiendo el orden en el que se han ubicado los eventos. Una vez se ha realizado la simulación y todos los eventos han sido ejecutados, se ha de destruir todos los objetos auxiliares que hayan sido creados, realizando así una limpieza del sistema. Esto se hace añadiendo la siguiente línea de código justo después de la anteriormente descrita (*run*).

```
Simulator::Destroy ();
```

Capítulo 7. Modelo NS-3 de Sigfox

Una de las tecnologías de comunicación con más protagonismo a día de hoy en el sector IoT es Sigfox, la cual está basada en el estándar de comunicación ETSI LTN. Gran parte del mercado se ha decidido por esta solución debido al gran nivel de implantación que tiene actualmente en relación con sus competidores, llegando a dar una cobertura del cien por cien a más de treinta países. En España, la red está operada por Cellnex Telecom y ya cuenta con un gran cliente a nivel nacional como es Securitas Direct.

Por ello, sumado al carácter libre del estándar, se ha desarrollado un modelo NS-3 para la simulación de redes Sigfox, permitiendo obtener información en cuanto al consumo de potencia, el alcance, la calidad del enlace o la tasa de errores.

7.1 Estructura

El modelo de simulación desarrollado para la virtualización de una red Sigfox se compone de los módulos indicados en el siguiente diagrama:

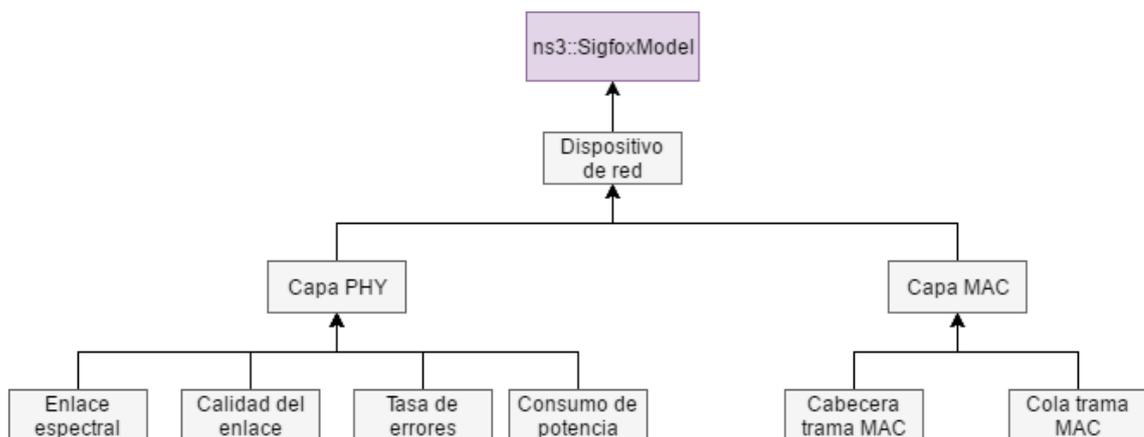


Figura 29 – Diagrama modelo Sigfox NS-3

- **enlace espectral:** modela las características del enlace de comunicación inalámbrico
- **calidad del enlace:** evalúa la calidad del enlace
- **tasa de errores:** evalúa la tasa de errores en la comunicación
- **consumo de potencia:** evalúa el consumo de potencia del dispositivo de red
- **capa PHY:** modela el funcionamiento de la capa física. Los cuatro módulos anteriores colaboran en realizar esta tarea
- **cabecera de la trama MAC:** permite crear la cabecera de los paquetes de comunicación
- **cola de la trama MAC:** permite crear la cola de los paquetes de comunicación
- **capa MAC:** modela el funcionamiento de la capa MAC. Se basa en los dos módulos anteriores para crear los paquetes de información
- **dispositivo de red:** permite la implementación de dispositivos Sigfox.

7.2 Módulos

Todos los módulos nombrados en el apartado anterior conforman el modelo de simulación de una red Sigfox. Cada uno de ellos tiene una relevancia de mayor o menor nivel, pero los principales son la capa PHY y la capa MAC. En este punto se va a detallar el funcionamiento de cada uno de ellos.

7.2.1 Capa PHY

La función de la capa física de OSI es la de codificar en señales los dígitos binarios que representan las tramas de comunicación, además de transmitir y recibir estas señales a través de los medios físicos. El objetivo de la capa física es crear la señal óptica, eléctrica o de microondas que representa a los bits en cada trama.

La capa física consiste en un hardware (conectores, medios y circuitos electrónicos), al contrario que las operaciones y los protocolos de las capas superiores que son por software. Su funcionamiento es similar al de una máquina de estados, de modo que actúa de una forma determinada dependiendo del momento en el que nos encontremos.

En concreto, nuestra capa PHY del modelo Sigfox ha sido diseñada como una máquina de estados finitos que sigue el diagrama de la *figura 30*. Los estados en los que puede encontrarse son:

- **sleep:** es el modo natural de funcionamiento del dispositivo. Su única función es estar en escucha por el canal inalámbrico hasta que se le indique un cambio de estado por parte de la capa superior. El nivel de consumo energético es mínimo ya que no realiza ninguna función.
- **transmisión (TX_ON):** se establece este modo de transmisión cuando se quiere enviar información a través del canal. Supone el estado con mayor consumo energético.
- **recepción (RX_ON):** se llega a este estado cuando se abre una ventana para recepción según marca el estándar (ver *punto 4.2.1.4*)
- **transient:** se establece un estado transitorio cada vez que se termina una transmisión o recepción.

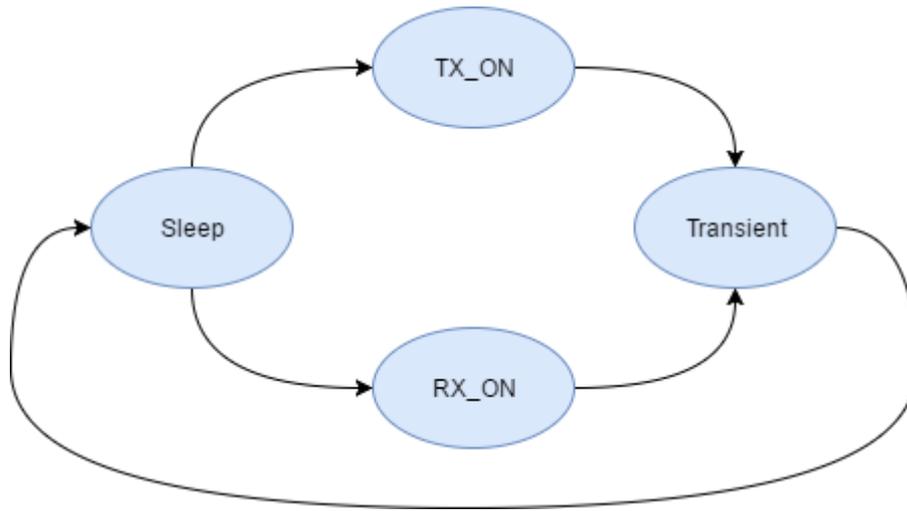


Figura 30 - Máquina de estados PHY

Además de establecerse la máquina de estados que gestiona la capa física, se instancian cuatro módulos para: la definición del enlace y sus características físicas, estudiar la calidad del mismo, evaluar la tasa de errores en la comunicación y obtener información sobre el consumo de potencia que tiene el dispositivo modelado.

7.2.1.1 Enlace espectral

Este módulo se encarga de establecer un modelo del espectro y definir dentro de él los canales correspondientes para ser usados por el dispositivo. Se crea un modelo espectral mediante una función propia de ns3 (*ns3::SpectrumModel*) en el que se representan 20000 bandas, definiéndose las frecuencias centrales, inferiores y superiores, dentro del rango de frecuencias de 868,0 – 868,60 MHz, abarcando un total de 200 kHz como define Sigfox.

Como siguiente paso para definir el espectro, que viene a ser un vector en el que cada posición es una banda de 10 Hz, se definen dentro de él los canales de comunicación, asignándoles un ancho de banda de 100 Hz (10 bandas) acorde al estándar.

Definido así el modelo del espectro, se crea a partir de él un modelo de densidad espectral, en el cual se le asigna a las distintas bandas del canal seleccionado para transmitir un valor de densidad espectral de potencia siguiendo el patrón de ancho de banda efectivo establecido por la tecnología UNB. A su vez, se define un modelo espectral de ruido para el mismo canal seleccionado, el cual se elige siguiente el protocolo de acceso al medio RFTDMA (ver *punto 4.2.1.1*)

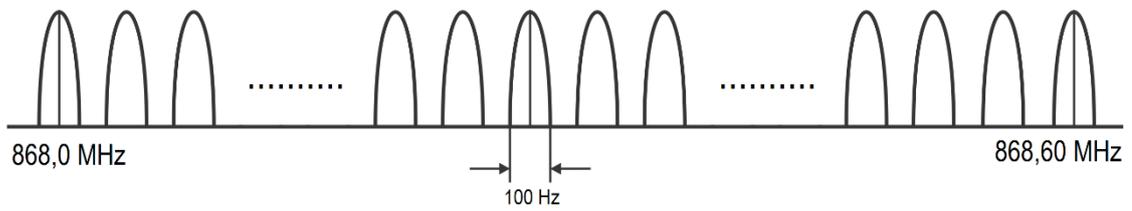


Figura 31 - Espectro del enlace

El modelo de dependencias de este módulo viene definido en el grafo de la figura siguiente:

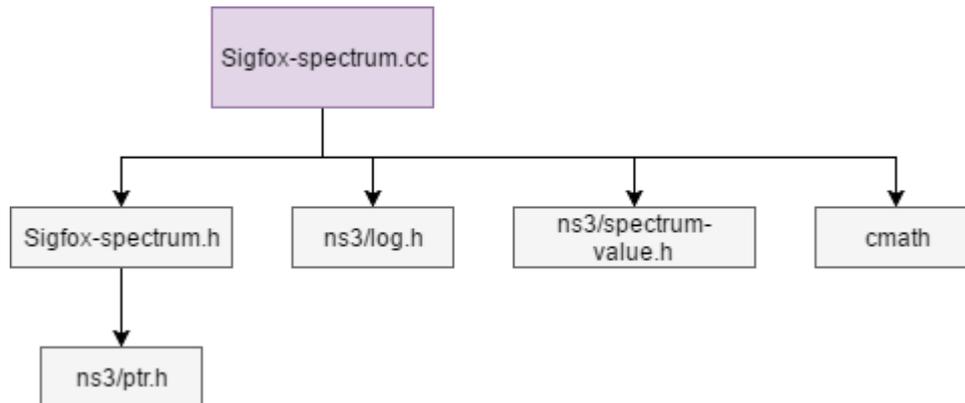


Figura 32 - Dependencias Sigfox-spectrum

7.2.1.2 Calidad del enlace

Mediante este módulo se define el indicador de calidad del enlace del último paquete recibido. Se establece según la intensidad de señal recibida y el número de errores. Su modelo de dependencias viene definido en el siguiente grafo:

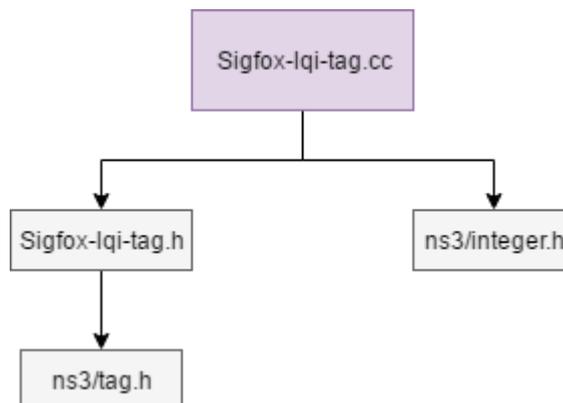


Figura 33 - Dependencias Sigfox-lqi

7.2.1.3 Tasa de errores

Este módulo para la simulación de una red Sigfox calcula cuando un paquete llega a destino o no. Este cálculo se realiza de acuerdo a la relación señal a ruido recibida, calculada a partir del enlace espectral y las pérdidas dadas por el modelo de propagación, y a una serie de coeficientes binomiales predefinidos que definen un modelo de errores base.

Además de calcular cuando un paquete se recibe correctamente, aporta información sobre la tasa de errores de transmisión.

Su grafo de dependencias viene definido en la siguiente figura:

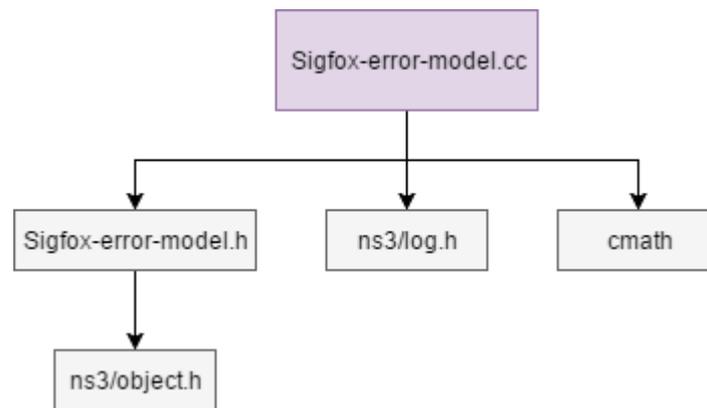


Figura 34 - Dependencias Sigfox-error-model

7.2.1.4 Consumo de potencia

Este módulo depende directamente de los distintos estados de la capa física. Cada vez que se realiza un cambio de uno de ellos a otro se realiza una interrupción avisando a este módulo del estado en el que nos encontramos.

El cálculo del consumo de potencia se realiza siguiendo las características eléctricas, en cuanto a tensión y corriente, de distintos chips que funcionan bajo la tecnología de comunicación Sigfox. En nuestro caso, se han establecido los valores de consumo de los chips Telit LE51-868S:

- **tensión de alimentación:** 3 V
- **consumo en recepción:** 32 mA
- **consumo en transmisión:** 55 mA
- **consumo en sleep:** 2 μ A

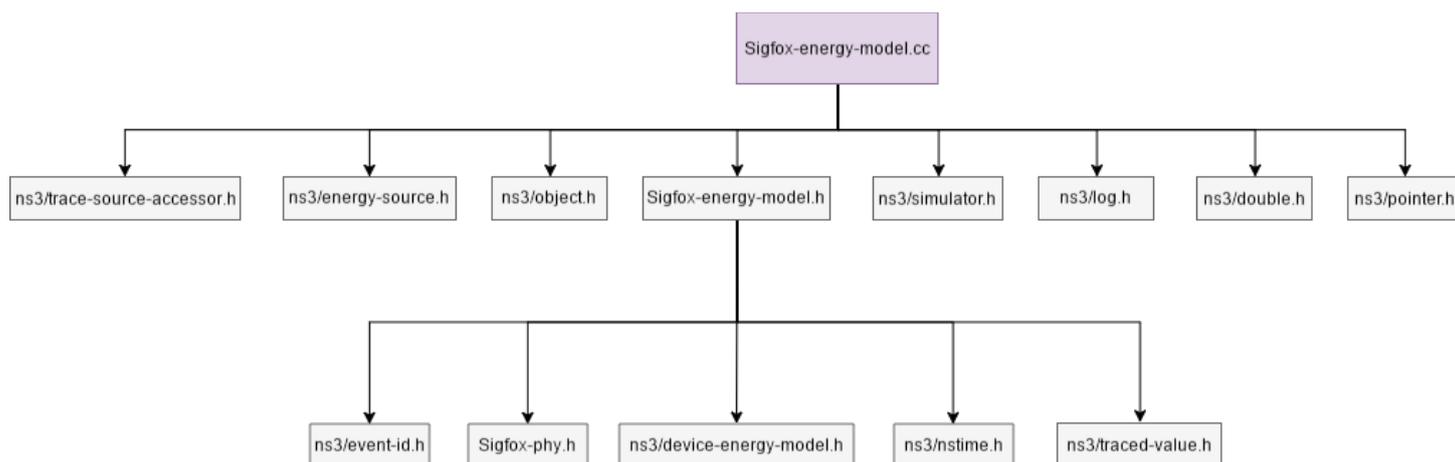


Figura 35 - Dependencias Sigfox-energy-model

7.2.2 Capa MAC

La capa de control de acceso al medio (MAC, (“*Medium Access Control*”) define una serie de funciones para realizar las operaciones propias de las redes inalámbricas. La capa MAC se encarga, en general, de gestionar y mantener las comunicaciones entre estaciones. La capa MAC tiene que coordinar el acceso a un canal de radio compartido y utilizar su capa física (PHY) para transmitir y recibir, así como realizar el montaje de las tramas.

El acceso al medio lo obtiene bajo el protocolo RFTDMA (ver *punto 4.2.1.1*), el cual define la selección de un canal de transmisión aleatorio sin preocuparse si el canal está ocupado o no.

Además de controlar el acceso al medio y formar los paquetes que van a ser transmitidos, la capa MAC se encargar de realizar el tratamiento de las tramas recibidas. En nuestra capa diseñada para una red Sigfox podemos trabajar en dos modos de funcionamiento: promiscuo o no promiscuo. En el primero se aceptan todos los paquetes recibidos sin importar ni verificar ninguno de los campos de la trama. En el segundo, tras detectarse por el canal de radio el preámbulo predefinido, se realizan dos niveles de filtrado antes de ser aceptadas las tramas:

1. Primeramente se verifica que la dirección destino de la trama coincide con la dirección del dispositivo
2. Y en segundo lugar se realiza un FCS de la cabecera y la carga del paquete, comparándose con el valor del campo FCS de la trama, el cual viene representado en la cola. Si ambos valores coinciden, el paquete es aceptado

De la misma forma que en la capa PHY se utilizan una serie de estados para controlar su funcionamiento, la capa MAC utiliza cuatro estados para su propio flujo de trabajo:

- **MAC_IDLE:** indica que la capa MAC no está trabajando. Está libre para realizar cualquier función, incluso recibir tramas de la capa física
- **MAC_SENDING:** indicia que la capa MAC está enviando la trama a la capa inferior (PHY)
- **MAC_FAILURE:** indica que ha ocurrido un problema en la capa MAC. Puede ser debido a un error de acceso al canal, montaje de las tramas o fallos derivados de la recepción de estas, como el no pasar uno de los filtros

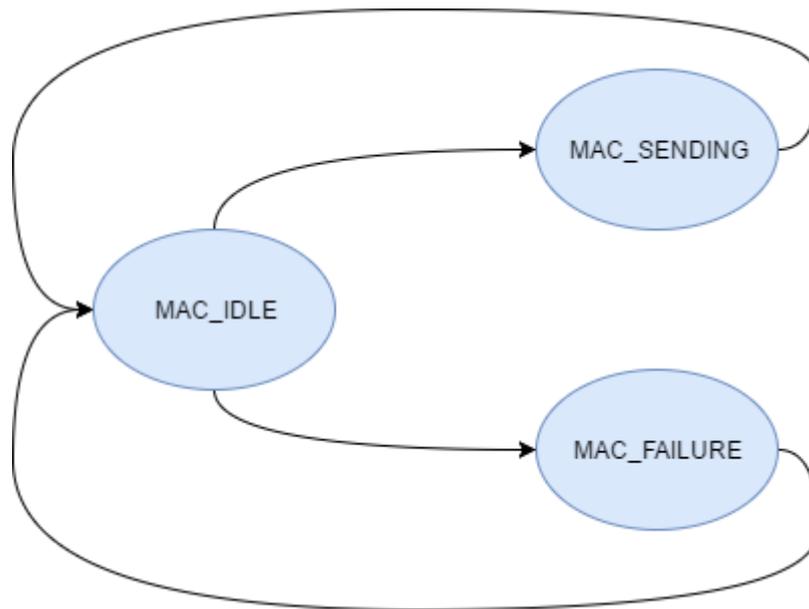


Figura 36 - Estados capa MAC

7.2.2.1 Cabecera de la trama MAC

Este módulo se encarga del montaje de la cabecera de la trama. Va ligada directamente al modelo paquete del simulador (*ns3::Packet*) y más concretamente al módulo cabecera (*ns3::header*), como podemos ver en el grafo de dependencia siguiente:

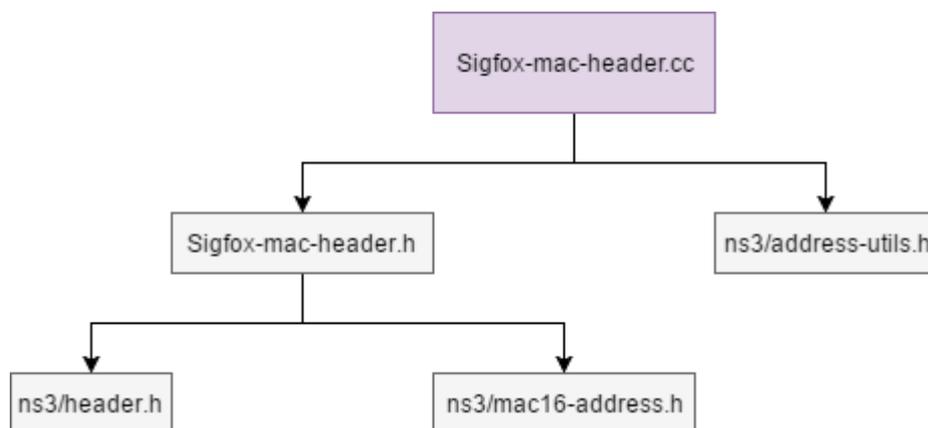


Figura 37 - Dependencias MAC header

Los campos de la trama que instancia este módulo son los que podemos ver en las ilustraciones 20 y 21, siendo el valor del campo ID del dispositivo destino una dirección MAC de 16 bits.

7.2.2.2 Cola de la trama MAC

Al igual que la cabecera, la cola de la trama MAC está ligada a el modelo paquete de NS-3. En este caso en concreto, va ligada al módulo cola del simulador (*ns3::trailer*), como podemos ver en el siguiente grafo:

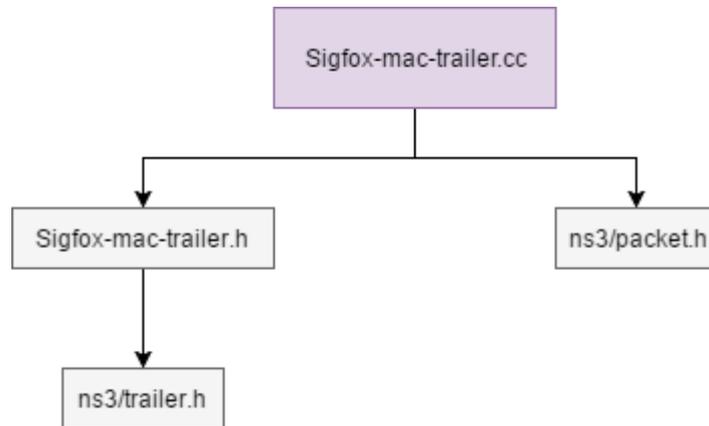


Figura 38 – Dependencias MAC trailer

Su función es realizar un cálculo FCS con el resto del paquete ya montado (cabecera y carga)

Capítulo 8. Estudios realizados

Con el modelo NS-3 para Sigfox diseñado en el presente trabajo, podemos programar scripts de simulación para estudiar tanto las características de una red Sigfox como las del propio estándar de comunicación. Por ello, se permite simular:

- el alcance de los enlaces de subida y bajada
- el contenido de las tramas enviadas y el manejo de ellas
- la calidad del enlace de comunicación
- la tasa de errores en la comunicación
- la aleatoriedad del protocolo de acceso al medio (RFTDMA) y por consiguiente la probabilidad de ocupación de los canales en el tiempo
- el consumo de potencia de los distintos dispositivos de la red con un ciclo de trabajo predefinido

De este modo, se han realizado dos scripts de simulación para obtener información sobre la cobertura en un enlace de subida (teóricamente el de menor alcance de los dos, al tener una potencia de transmisión menor) y el consumo de potencia de un dispositivo Sigfox durante un ciclo de trabajo establecido.

Además, como valor añadido, se ha realizado un estudio del protocolo RFTDMA para dejar constancia de la reducida probabilidad de que ocurran colisiones en una red.

8.1 Estudio de alcance

En este estudio se ha realizado un script que nos permite obtener el número de paquetes que llegan a un nodo destino desde un nodo transmisor en diferentes entornos (rural, suburbano y urbano) y colocados a la misma altura, con una determinada potencia de transmisión y para un cierto tamaño de paquete. A su vez, se establece el envío de una gran cantidad de estos paquetes cada cierta distancia, la cual va incrementándose hasta un límite fijado.

De esta forma, las características de la simulación son las siguientes:

- **potencia de transmisión:** 14 dBm (25 mW)
- **distancia inicial:** 0 m
- **distancia máxima:** 25 km
- **incremento de distancia:** 200 m
- **tamaño del paquete:** 20 bytes
- **número de paquetes enviados:** 1000 paquetes en cada distancia de separación
- **sensibilidad del receptor:** -130 dBm
- **modelo de propagación:** OkumuraHata
- **entornos de simulación:** rural, suburbano y urbano

El modelo de propagación Okumura-Hata es uno de los más utilizados para la predicción de pérdidas en el rango de 100 MHz a 1920 MHz y está basado en una extensa batería de medidas tomadas durante un proyecto llevado a cabo en Tokio. El principal resultado de este trabajo fue

un conjunto de curvas que proporcionan el nivel de atenuación media relativa al espacio libre, en función de la frecuencia, la distancia entre transmisor y receptor, la altura de las antenas de la estación base y la estación móvil, además de varios factores de corrección específicos para diferentes tipos de trayecto.

Con el objetivo de hacer que este método fuera más fácil de aplicar, se estableció una serie de relaciones numéricas que describen el método gráfico propuesto, dando lugar a un modelo de propagación. Este modelo está considerado entre los más simples y mejores en términos de su precisión en el cálculo de las pérdidas en el trayecto y se ha convertido en el método de planificación de sistemas móviles en Japón.

El modelo Okumura-Hata expresa la pérdida básica de propagación de la siguiente forma:

$$L_b = 69,55 + 26,16 \log f - 13,82 \log h_b - a(h_m) + (44,9 - 6,55 \log h_b) \log d_m$$

Figura 39 - Pérdidas básicas Okumura-Hata

Donde $a(h_m)$ es un factor de corrección que depende de la altura de la antena y que se calcula para distintos entornos de la siguiente manera:

1. Áreas rurales:

$$L_b = L_b(\text{urban}) - 4,78 \log(f)^2 + 18,33 \log f - 40,94$$

Figura 40 - Factor de corrección rural

2. Áreas suburbanas

$$L_b = L_b(\text{urban}) - 2[\log(f/28)]^2 - 5,4$$

Figura 41 - Factor de corrección suburbano

3. Áreas urbanas

- Ciudades pequeñas o medianas ($1 \leq h_m \leq 10$ m)

$$a(h_m) = (1,1 \log f - 0,7)h_m - (1,56 \log f - 0,8)$$

Figura 42 - Factor de corrección ciudades pequeñas

- Ciudades grandes

$$a(h_m) = \begin{cases} 8,29(\log 1,54h_m)^2 - 1,1, & f \leq 200 \text{ MHz} \\ 3,2(\log 11,75h_m)^2 - 4,97, & f \geq 400 \text{ MHz} \end{cases}$$

Figura 43 - Factor de corrección ciudades grandes

En los siguientes puntos podemos ver las simulaciones realizadas para el estudio del alcance en los tres entornos nombrados:

8.1.1 Entorno rural

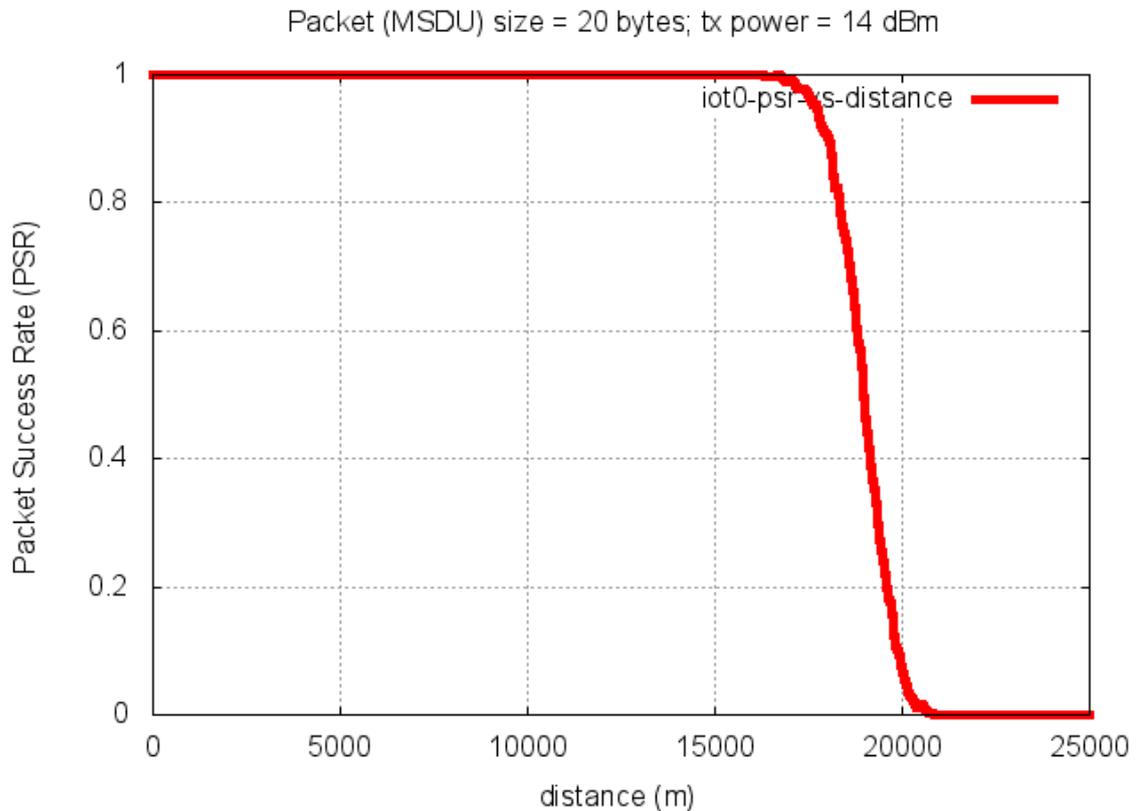


Figura 44 - Escenario rural

En los resultados obtenidos en la simulación de la figura anterior podemos observar como el éxito de envío en los paquetes se mantiene en el 100% hasta una distancia de 17 km, y llega al 90% de éxito sobre los 18 km. De esta forma, se puede determinar que una comunicación Sigfox fiable llegaría hasta los 18 km en entornos rurales.

Los resultados obtenidos en este ambiente de simulación son algo dispares de los proporcionados por Sigfox (ver *tabla 1*). No obstante, hay un detalle importante en cuanto a la potencia de transmisión que dan sentido a nuestros resultados. En los datos proporcionados por la empresa comercializadora de esta tecnología expresan que la potencia de transmisión puede llegar hasta los 100 mW, pero no indican con que potencia ha sido realizada la estimación de alcance. En nuestro caso, esta potencia está fijada a 25 mW, apenas un cuarto de la máxima aportada, ya que la regulación en España, y muchos otros países, establecen el máximo de transmisión en la banda 868 MHz a este valor de potencia.

De esta forma, nuestros resultados serían aceptables, llegando a una distancia de 18 km con una potencia de transmisión de 14 dBm.

8.1.2 Entorno suburbano

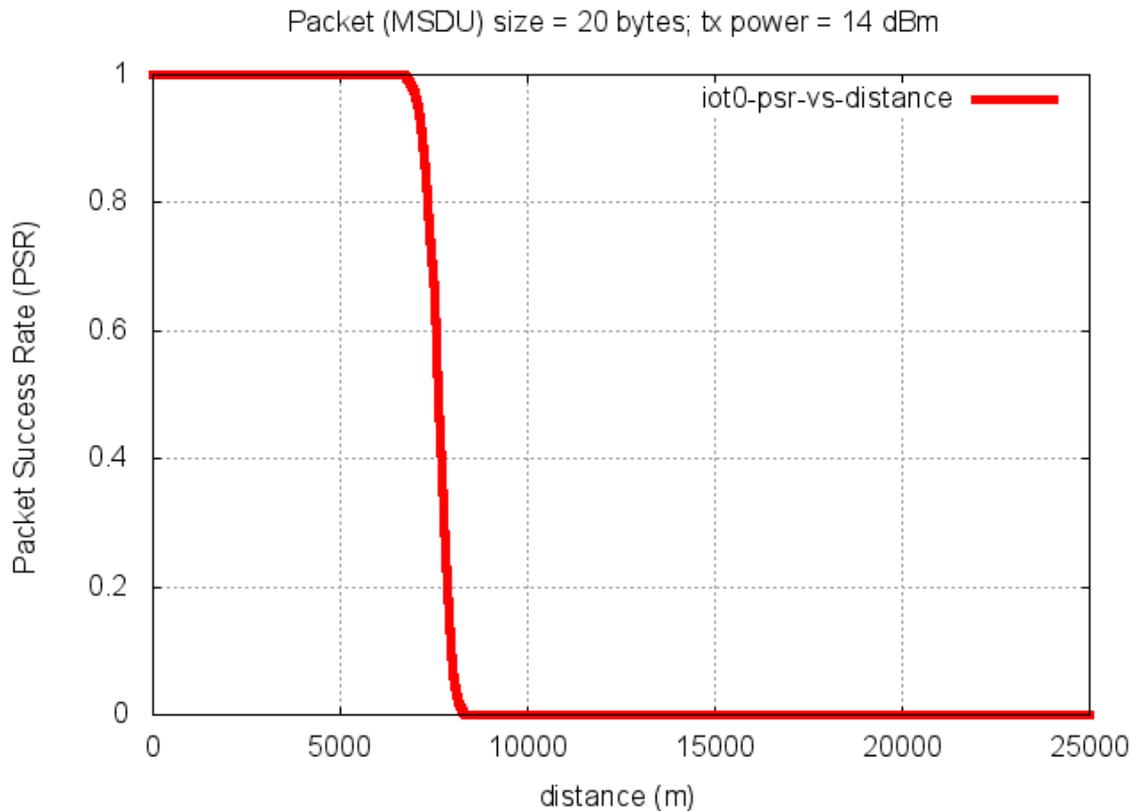


Figura 45 - Escenario suburbano

En esta simulación, al igual que en la anterior, tenemos el mismo problema en los criterios de potencia de transmisión. La *tabla 1* establece un alcance de hasta 10 km sin diferenciar entornos urbanos o suburbanos y, nuevamente, sin dar un valor de potencia de transmisión con el que referenciar nuestros resultados.

De este modo, los valores obtenidos en nuestra simulación para una potencia de transmisión de 25 mW no son muy dispares de los aportados por Sigfox. Alcanzamos distancias de hasta 7,5 km con un 90% de acierto en la tasa de envío de paquetes.

8.1.3 Entorno urbano

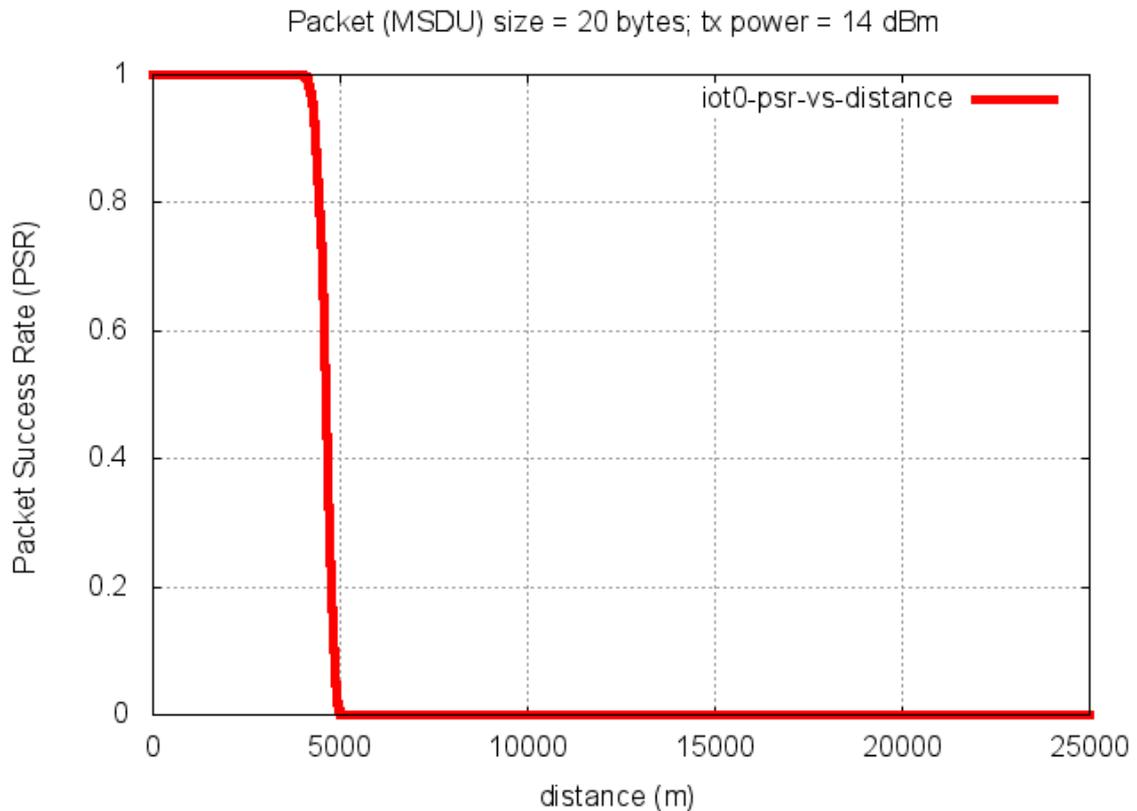


Figura 46 - Escenario urbano

Por último, en el entorno urbano tenemos de nuevo la misma problemática con los datos proporcionados por Sigfox. En este caso, llegamos a distancias de hasta 4 km con una tasa de acierto del 90%.

8.2 Estudio de consumo de potencia

En este estudio se ha realizado un script para calcular el consumo de potencia de un chip Telit LE51-868S, usando sus características eléctricas:

- **tensión de alimentación:** 3 V
- **consumo en recepción:** 32 mA
- **consumo en transmisión:** 55 mA
- **consumo en sleep:** 2 μ A

Se ha establecido un ciclo de trabajo con una duración de un minuto. En primer lugar, se realiza una transmisión (TX_ON) de un paquete de 12 bytes, la cual se efectúa en 1,6 segundos (12 bytes a 100 bps). Una vez realizado el envío de información se pasa a un estado de trabajo reducido (sleep) tras el cual se abre una ventana de recepción y pasamos al estado RX_ON. Tras

finalizar este estado en el que podemos recibir información y cuya duración es de 5 segundos, volvemos a estar en modo sleep hasta el fin del ciclo de trabajo.

Cabe comentar que, como ya se ha dicho anteriormente en el *punto 7.2.1*, se produce un estado transitorio al finalizar cada estado de TX_ON y RX_ON.

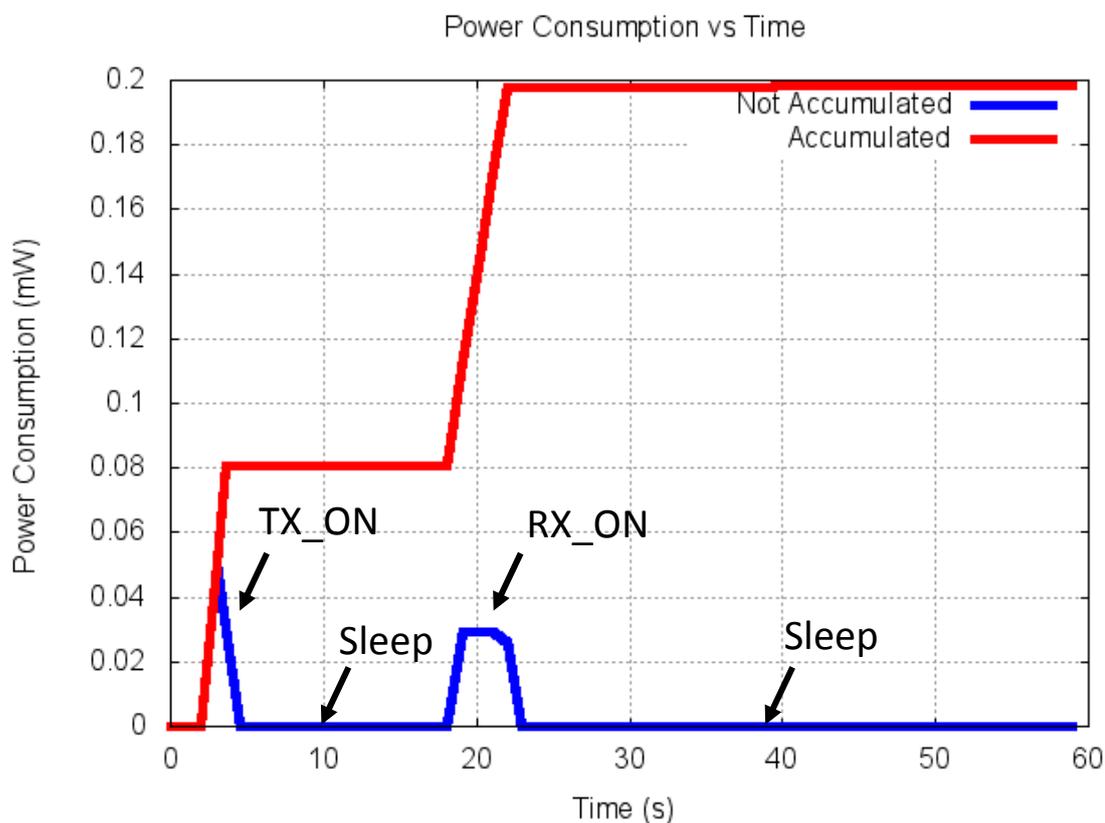


Figura 47 - Consumo de potencia

8.3 Estudio del método de acceso al medio (RFTDMA)

El objetivo de este estudio es indicar la baja probabilidad de colisión de paquetes que hay en la tecnología Sigfox, ya que al no tener un método de acceso al medio que tenga en cuenta la ocupación del canal, puede llegarse a pensar que no es efectiva.

Esta tecnología habilita 2000 canales a lo largo de 200 KHz con un ancho de banda de cada uno de ellos de unos 100 Hz. A lo largo de un día, cada nodo puede realizar el envío de 140 mensajes de un tamaño de 12 bytes. Lo que se traduce en 140 ocupaciones del canal en 24 horas, con una duración de esta ocupación de 1,6 segundos. En total, el canal está ocupado 224 segundos al día (3 minutos y 44 segundos).

En nuestro script se ha diseñado el envío de 1000 paquetes consecutivos por parte de un mismo nodo para observar la distribución en la elección de los canales. La *figura 48* muestra estos resultados y se puede observar que la posibilidad de repetición del uso de un canal es muy

reducida. De este modo, si varios nodos transmitieran en un mismo instante de tiempo, algo que en una red Sigfox real sería poco probable, la probabilidad de que uno de estos elija el mismo canal de transmisión que otro nodo de la misma red es altamente improbable.

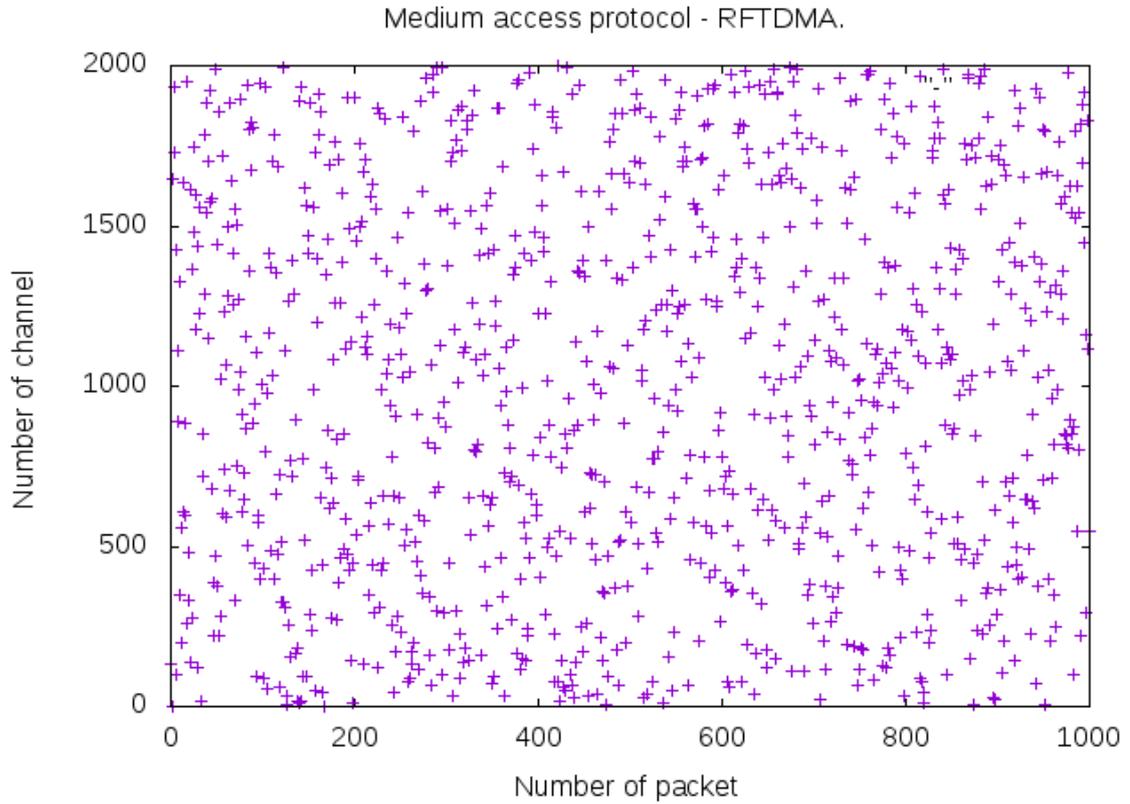


Figura 48 - Simulación RFTDMA

Capítulo 9. Conclusiones y trabajo futuro

Como se ha fundamentado a lo largo del presente trabajo, IoT es una de las grandes apuestas tecnológicas de hoy en día. Es por ello que el estudio de las tecnologías de soporte para IoT adquiere especial relevancia, más todavía teniendo en cuenta la disparidad de enfoques (público frente a propietario, radio de cobertura frente a ancho de banda, y consumo frente a ambas) que requieren el amplio abanico de aplicaciones IoT.

A la hora realizar un proyecto para instalar una red de comunicación inalámbrica hay varios factores que son muy importantes. Entre ellos destaca por un lado la cobertura que puede ofrecer dicha red, ya que determinará el número de estaciones base que habrá que instalar para dar cobertura a una determinada región. Y por otro lado destaca el consumo energético que pueden tener los distintos dispositivos que forman parte de la propia red, ya que en aplicaciones típicas del sector IoT es esencial tener dispositivos que funcionen en unos niveles de potencia reducidos, llegando a tener vidas útiles de entorno a decenas de años.

Los primeros capítulos de este trabajo están dedicados a proporcionar una panorámica general acerca de algunas de estas tecnologías, sistematizando su análisis y proporcionando una tabla-resumen de características necesarias para elegir la mejor opción para una aplicación dada.

Los siguientes capítulos se centran en el estudio de dos de las propuestas más interesantes: LoRa y Sigfox. Para LoRa se ha optado por un enfoque eminentemente práctico, realizando baterías de pruebas para contrastar las especificaciones ofertadas por el fabricante con las prestaciones reales de los dispositivos, llegando a la conclusión de que las especificaciones son, en la mayoría de los casos, muy optimistas respecto a la realidad.

El segundo caso de estudio se ha basado en técnicas de simulación. Hoy en día las simulaciones son muy importantes en distintos ámbitos, tanto académicos como empresariales. Por la vía investigadora, dentro del marco académico, permiten el estudio de una gran variedad de escenarios con diferentes características, lo que lleva a la posibilidad de desarrollar mejoras o proponer alternativas. En el ámbito empresarial coge una importancia mayor si cabe. Ofrece a las empresas la posibilidad de simular sus proyectos y realizar los pertinentes cálculos de interés, ofreciéndose así un ahorro económico importante, ya que evita que se tenga que realizar la implementación de dichos proyectos sin saber sus prestaciones reales.

Por ello, para este segundo estudio se ha implementado un modelo de simulación en entorno NS-3 de la tecnología de comunicación Sigfox con el objetivo de evaluar las prestaciones descritas en el *capítulo 8*, centrándose especialmente en la cobertura de la red y el consumo de potencia en diferentes entornos.

Los resultados de estos estudios diferentes se presentan en los *puntos 8.1* y *8.2*. Éstos corroboran las especificaciones aportadas por Sigfox, puesto que se cumplen totalmente las prestaciones reflejadas en la documentación aportada por los desarrolladores de la tecnología. De forma indirecta, este hecho corrobora que se puede considerar que el modelo de simulación diseñado es representativo con el comportamiento de esta tecnología de comunicación.

El estudio realizado, así como el exhaustivo modelo de simulación desarrollado, ha sido aceptado y expuesto en formato poster en el WIICT, organizado por ITACA, el cual es un foro de discusión acerca de los nuevos avances en el campo de las TIC.

Además, el estudio abre las puertas a un amplio abanico de trabajos futuros, como puede ser la modelización de otros escenarios de simulación donde evaluar las prestaciones de Sigfox.

Asimismo, abre la puerta para introducir nuevas mejoras y parámetros de comunicación en la tecnología Sigfox

Se plantea como trabajo futuro el uso del modelo de simulación para la evaluación comparativa con otras tecnologías, como Zigbee, LTE o WiFi, siendo éstas tecnologías tradicionales que se aplican en ciertos ámbitos del sector IoT.

Se contempla, además, la validación de todos los modelos de simulación anteriores mediante medidas obtenidas con implementaciones reales de estas tecnologías en entornos reales, tal como se ha realizado con LoRa. Ello permitirá verificar los resultados obtenidos de las simulaciones del modelo NS-3 en los distintos entornos (rurales, suburbanos y urbanos).

Finalmente, con toda la experiencia acumulada de estos estudios se pretende proponer una nueva tecnología de soporte IoT que balancee los tres factores fundamentales (cobertura, ancho de banda y consumo) para aplicaciones IoT en función de los requisitos de las mismas. Dicha tecnología sería objeto de la Tesis Doctoral del autor.

Bibliografía

- [1] EOI - Escuela de Organización Industrial, "Las tecnologías IoT dentro de la industria conectada 4.0"
Disponible en: <https://www.eoi.es/es/file/20417/download?token=FSUSqFIt>
- [2] Brian Ray, 2017, LTE-M & 2 Other 3GPP IoT Technologies To Get Familiar With
Disponible en: <https://www.link-labs.com/lte-iot-technologies/>
- [3] Low Throughput Networks (LTN); Use Cases for Low Throughput Networks.
Disponible en: http://www.etsi.org/deliver/etsi_gs/LTN/001_099/001/01.01.01_60/gs_LTN001v010101p.pdf
- [4] Low Throughput Networks (LTN); Functional Architecture.
Disponible en: http://www.etsi.org/deliver/etsi_gs/LTN/001_099/002/01.01.01_60/gs_LTN002v010101p.pdf
- [5] Low Throughput Networks (LTN); Protocols and Interfaces.
Disponible en: http://www.etsi.org/deliver/etsi_gs/LTN/001_099/003/01.01.01_60/gs_ltn003v010101p.pdf
- [6] <https://www.sigfox.com/es>
- [7] <https://www.lora-alliance.org/>
- [8] <http://www.weightless.org/>
- [9] <http://www.ingenu.com/technology/rpma/>
- [10] <http://www.dash7-alliance.org/>
- [11] <http://www.nwave.io/>
- [12] Jonas Olsson, 6LoWPAN demystified.
Disponible en: <http://www.ti.com/lit/wp/swry013/swry013.pdf>
- [13] Isaac Brown, A Detailed Breakdown of LPWAN Technologies and Providers,
Disponible en: http://web.luxresearchinc.com/hubfs/Insight_Breakdown_of_LPWAN_Technologies.pdf?t=1461874447328
- [14] <http://lpwanforum.com/>
- [15] <http://dgmatics.com/about-waviot-company>
- [16] <https://iot-daily.com/>
- [17] <http://www.postscapes.com/long-range-wireless-iot-protocol-lora/>
- [18] U. Raza, P. Kulkarni and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," in IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 855-873, Secondquarter 2017.
- [19] http://wiki.dragino.com/index.php?title=Lora_Shield
- [20] Weiping Sun, Munhwan Choi and Sunghyun Choi, "IEEE 802.11ah: A Long Range 802.11 WLAN at sub GHz"
Disponible en: <http://www.mwnl.snu.ac.kr/~schoi/publication/Journals/13-ICT-SUN-rev.pdf>

- [21] Orden IET/787/2013, de 25 de abril, por la que se aprueba el cuadro nacional de atribución de frecuencias.
Disponible en: <http://www.boe.es/boe/dias/2013/05/09/pdfs/BOE-A-2013-4845.pdf>
- [22] Claire Goursaud, Jean-Marie Gorce. Dedicated networks for IoT : PHY / MAC state of the art and challenges. EAI endorsed transactions on Internet of Things, 2015, <10.4108/eai.26-10-2015.150597>. <hal-01231221>
- [23] Telit LE51-868S datasheet.
Disponible en: http://www.telit.com/fileadmin/user_upload/media/products/short_range/LE51-868-S/Telit_LE51-868_S_Datasheet.pdf
- [24] <https://www.xirio-online.com/help/es/okumura-hata.htm>

Anexo - Acrónimos

| | |
|-------|---|
| ADSL | <i>Asymmetric Digital Subscriber Line</i> |
| API | <i>Application Program Interface</i> |
| bps | <i>bits per second</i> |
| BPSK | <i>Binary Phase-Shift Keying</i> |
| BSS | <i>Business Support System</i> |
| CoAP | <i>Constrained Application Protocol</i> |
| CRA | <i>Central Registration Authority</i> |
| CRC | <i>Cyclic Redundancy Check.</i> |
| DBPSK | <i>Differential Binary Phase-Shift Keying</i> |
| DSSS | <i>Direct Sequence Spread Spectrum</i> |
| FCS | <i>Frame Check Sequence</i> |
| FTDMA | <i>Frequency-Time Division Multiple Access</i> |
| GFSK | <i>Gaussian Frequency Shift-Keying</i> |
| GMSK | <i>Gaussian Minimum Shift Keying</i> |
| GPRS | <i>General Packet Radio Service</i> |
| HTTP | <i>HyperText Transfer Protocol</i> |
| IoT | <i>Internet of Things</i> |
| IP | <i>Internet Protocol</i> |
| ITACA | <i>Instituto Universitario de Investigación en Aplicaciones de las Tecnologías de la Información y las Comunicaciones Avanzadas</i> |
| kcps | <i>kilo chip per second</i> |
| LAP | <i>LTN Access Point</i> |
| LEP | <i>LTN End Point</i> |
| LTN | <i>Low Throughput Network</i> |
| LPWAN | <i>Low-Power Wide-Area Network</i> |
| MAC | <i>Medium Access Control</i> |
| MQTT | <i>Message Queue Telemetry Transport</i> |
| OFDM | <i>Orthogonal Frequency Division Multiplexing</i> |
| OSS | <i>Operation Support System</i> |
| OSSS | <i>Orthogonal Sequence Spread Spectrum</i> |
| PHY | <i>PHYsical layer</i> |

| | |
|--------|--|
| REST | <i>REpresentational State Transfer</i> |
| RFTDMA | <i>Random Frequency and Time Division Multiple Access</i> |
| RX | <i>Reception</i> |
| TCP | <i>Transmission Control Protocol</i> |
| TDMA | <i>Time Division Multiple Access</i> |
| TIC | <i>Tecnologías de la Información y la Comunicación</i> |
| TX | <i>Transmission</i> |
| UDP | <i>User Datagram Protocol</i> |
| UNB | <i>Ultra Narrow Band</i> |
| WAN | <i>Wide Area Network</i> |
| WIICT | <i>Workshop on Innovation on Information and Communications Technologies</i> |