



**Universidad**  
Zaragoza



## Trabajo Fin de Máster

# Desarrollo de una Red de Sensores Inalámbrica Mallada para Gestión Remota de Alumbrado Público

Autor

Miguel Sousa Girón

Director/es

José Ramón Gállego Martínez

Universidad de Zaragoza, Escuela de Ingeniería y Arquitectura

2018





**DECLARACIÓN DE  
AUTORÍA Y ORIGINALIDAD**

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. Miguel Sousa Girón

con nº de DNI 72084561P en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Máster \_\_\_\_\_, (Título del Trabajo)

Desarrollo de una red de sensores inalámbrica mallada para gestión remota de alumbrado público

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 14 de Septiembre de 2018

Fdo: Miguel Sousa Girón



# Desarrollo de una Red de Sensores Inalámbrica Mallada para Gestión Remota de Alumbrado Público

## Resumen

El auge de los dispositivos conectados a Internet y la clara apuesta de instituciones públicas así como de empresas por un tipo de desarrollo urbano basado en la sostenibilidad, convergen en un amplio abanico de soluciones en el mercado donde las redes de comunicaciones y los sistemas de telecomunicación conforman un pilar fundamental.

El presente Trabajo Fin de Máster presenta el estudio, diseño e implementación de una red de dispositivos mallada e inalámbrica sustentada por estándares y protocolos abiertos que permiten desarrollar un sistema compatible y adaptable para la gestión y el traspaso de información correspondiente a distintos servicios urbanos como el alumbrado público, el agua o la energía. La red de comunicaciones desarrollada formará parte del proyecto STELARIA, un sistema de gestión remoto e inalámbrico punto a punto para alumbrado público, desarrollado por la empresa española ELT (Especialidades Luminotécnicas S.A.).

En primer lugar, se realiza un análisis del estado del arte correspondiente a las ciudades inteligentes y los sistemas para gestión de alumbrado público que permiten describir el contexto que rodea a este proyecto. Después de analizar la evolución y el potencial obtenido tras proporcionar conectividad y acceso a Internet a los objetos cotidianos del día a día, se observa la inexistencia de un conjunto de protocolos que permitan estandarizar este tipo de comunicaciones. Por ello, en este trabajo se plantea la puesta en marcha de una red de comunicaciones basada en radiofrecuencia haciendo uso de estándares y protocolos abiertos.

Una vez definidos los requisitos y las especificaciones técnicas y funcionales de la red a implementar, se evalúa la integración de la red con el sistema STELARIA. Se valida su comportamiento en un despliegue sobre instalación real y por último, se realiza un análisis de los resultados obtenidos con el fin de validar las premisas iniciales.

La solución mostrada es una de las posibles alternativas que resuelve el problema de realizar una interconexión de dispositivos. Podemos encontrar un conjunto de soluciones alternativas que quedarán contempladas a lo largo de este trabajo así como al final de la memoria, donde se describen las líneas de trabajo futuro.



# Índice de Contenidos

---

<b>1. Introducción.....</b>	<b>17</b>
1.1. Antecedentes .....	17
1.1.1 Ciudades y Alumbrado Inteligente .....	17
1.1.2 Marco General del Proyecto .....	19
1.2. Objetivos y Alcance .....	20
1.2.1 Cronograma .....	21
1.3. Estructura de la Memoria.....	21
<b>2. Selección de Protocolos .....</b>	<b>23</b>
2.1. Nivel Físico y Nivel de Enlace. Selección de la Tecnología .....	23
2.1.1. Tecnologías Presentes en el Mercado.....	23
2.1.2. Criterios de Selección y Comparativa .....	26
2.2. Nivel de Red. ....	27
2.2.1. Protocolo de Red. IPv6 sobre 6LoWPAN .....	27
2.2.2. Protocolo de Enrutamiento. RPL.....	28
2.3. Nivel de Transporte. UDP .....	29
2.4. Nivel de Aplicación. CoAP .....	30
<b>3. Implementación de la Red de Comunicaciones.....</b>	<b>33</b>
3.1. Tipo de Dispositivos .....	33
3.2. Sistema Operativo. Contiki OS.....	34
3.3. Selección de Dispositivo Embebido .....	35
3.3.1. Requisitos .....	35
3.3.2. Selección de Arquitectura y Familia de Microcontrolador .....	35
3.4. Emulación del Diseño .....	37
3.4.1. Configuración de Contiki OS .....	37
3.4.2. Entorno de Emulación Cooja.....	38
3.5. Diseño Electrónico.....	40
3.5.1. Fuente de Alimentación.....	40
3.5.2. Placa Conexión 2G/3G .....	41
3.5.3. Placa Microcontrolador y Transceptor Radio.....	42
3.6. Procesos de Fabricación y Test.....	43
3.6.1. Proceso de Fabricación.....	43
3.6.2. Proceso de Test. Gestión de Fabricación Mediante ERP .....	44

3.6.3. Proceso de Test. Diseño de Herramienta Estadística .....	46
3.7. Pruebas y Validación .....	47
3.7.1. Validación del Firmware sobre Microcontrolador .....	47
3.7.2. Validación del Transceptor.....	48
3.8. Implementación del Protocolo de Aplicación.....	50
3.8.1. Implementación de Recursos CoAP.....	51
3.9. Despliegue de Validación .....	52
<b>4. Despliegue de la Red de Comunicaciones .....</b>	<b>57</b>
4.1. Despliegue de la Red en Instalación .....	57
4.2. Verificación del Despliegue.....	58
4.3. Beneficios del Despliegue y Estimación de Ahorro .....	60
<b>5. Conclusiones y Líneas de Trabajo Futuro .....</b>	<b>63</b>
5.1. Conclusiones .....	63
5.2. Líneas de Trabajo Futuro .....	64
<b>ANEXO A. Estándar IEEE 802.15.4 .....</b>	<b>67</b>
<b>ANEXO B. Bluetooth Low Energy .....</b>	<b>71</b>
<b>ANEXO C. Algoritmo RPL.....</b>	<b>74</b>
<b>ANEXO D. Sistema Operativo Contiki OS .....</b>	<b>75</b>
<b>ANEXO E. Código Aplicación Sniffer .....</b>	<b>89</b>
<b>ANEXO F. Estudio y Simulación del Radioenlace .....</b>	<b>90</b>
<b>ANEXO G. Sistema STELARIA .....</b>	<b>98</b>
<b>ANEXO H. Hojas de Características .....</b>	<b>107</b>
<b>ANEXO I. Capturas de Tráfico en Wireshark.....</b>	<b>109</b>
<b>Referencias .....</b>	<b>112</b>

## Lista de Símbolos y Acrónimos

---

6LoWPAN	IPv6 over Low Power Wireless Personal Area Networks
ADC	Analog - Digital Converter
AES	Advanced Encryption Standard
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BLE	Bluetooth Low Energy
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CoAP	Constrained Application Protocol
DAC	Digital - Analog Converter
DALI	Digital Addressable Lighting Interface
DEP	Densidad Espectral de Potencia
DNS	Domain Name System
eSIM	Embedded SIM
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute
ETX	Expected Transmisión Count
FTP	File Transfer Protocol
FOTA	Firmware Over the Air
LED	Light Emitting Diode
HCI	Host Controller Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IETF	Internet Engineering Task Force
I <sup>2</sup> C	Inter-Integrated Circuit
IoT	Internet of Things
JAAS	Java Authentication and Authorization Service
L2CAP	Logic Link Control and Adaptation Protocol

LDAP	Lightweight Directory Access Protocol
LQI	Link Quality Indicator
LR-WPAN	Low Rate Wireless Personal Area Network
LWM2M	Lightweight Machine To Machine.
MAC	Medium Access Control
MPDU	MAC Protocol Data Unit
NBIIoT	Narrow Band IoT
NLOS	Non Line of Sight
OSPF	Open Shortest Path First
PLC	Power Line Communications
PPDU	Physical Protocol Data Unit
RAM	Random Access Memory
RDC	Radio Duty Cycling
RIP	Routing Information Protocol
RISC	Reduced Instruction Set Computer
RPL	Routing over Low Power Lossy Networks
RSSI	Received Signal Strength Indicator
RTOS	Real-Time Operating System
SNR	Signal to Noise Ratio
SOA	Service Oriented Architecture
SoC	System on Chip
SPI	Serial Peripheral Interface
SRD	Short Range Devices
SSL	Solid State Lighting
TIC	Tecnologías de Información y Comunicación
UART	Universal Asynchronous Receiver-Transmitter
VSAP	Vapor de Sodio a Alta Presión
WSN	Wireless Sensor Network

## Lista de Figuras

---

Figura 1 Previsiones de evolución de la tecnología LED.....	17
Figura 2 Esquema de red de comunicaciones a través del alumbrado público .....	18
Figura 3 Bluetooth 4.0. Bluetooth Smart o Bluetooth Low Energy .....	23
Figura 4 Estándar 802.11. Wi-Fi .....	24
Figura 5 Esquema de red SIGFOX. Solución cerrada con acceso mediante API .....	24
Figura 6 Dispositivo LoRaWAN. Conexión con Servidor.....	25
Figura 7 Distintas implementaciones haciendo uso de estándar IEEE 802.15.4.....	25
Figura 8 Ejemplo de red 6LoWPAN .....	27
Figura 9 Comparativa precio vs escalabilidad. Tendencia del mercado .....	28
Figura 10 DODAG, Destination Oriented Directed Acyclic Graph.....	29
Figura 11 Comparativa cabeceras TCP y UDP .....	30
Figura 12 Intercambio mensajes entre cliente y servidor CoAP .....	31
Figura 13 Cabecera CoAP .....	31
Figura 14 Pila de protocolos obtenida para diseño de red de comunicaciones .....	31
Figura 15 Ejemplo de red WSN .....	34
Figura 16 Ejemplos de plataformas hardware que soportan Contiki OS .....	34
Figura 17 Familias de dispositivos para arquitectura ARM.....	36
Figura 18 Asignación de protocolos en capa TCP/IP del dispositivo. Contiki OS .....	37
Figura 19 Asignación a nivel de driver para pila TCP/IP. Contiki OS .....	38
Figura 20 Ejemplo de red simulada. 10 nodos simples y 1 nodo concentrador .....	39
Figura 21 Ejemplo de intercambio de mensajes entre nodos 10 y 1 a través de nodo 4	39
Figura 22 Representación 3D de PCB1. Fuente de alimentación.....	40
Figura 23 Representación 3D de la PCB2. Conexión 2G-3G .....	41
Figura 24 Representación 3D de la PCB3. Microcontrolador y transceptor radio.....	42
Figura 25 Dispositivo final cubierto por carcasa y enresinado .....	43
Figura 26 PCB2. Chipset 2G/3G y antena formato chip .....	44
Figura 27 Escáner rayos X sobre chipset 2G/3G. Burbujas en la soldadura de los pads	44
Figura 28 Útil diseñado para grabado y test de PCB3.....	45
Figura 29 Ejemplos sobre ERP de grabado y test de PCB .....	45
Figura 30 Resultados estadísticos de potencia y tiempo de test para PCB3 .....	46
Figura 31 Resultados de tensión de salida para fuente de alimentación de 5V. PCB1 ..	47
Figura 32 Código vinculación de fuentes de reloj entre microcontrolador y S.O.....	48
Figura 33 Vista 3D de la radiación. Plano X-Y.....	49
Figura 34 Vista 3D. Diagrama de radiación. Cortes del diagrama.....	49
Figura 35 Curvas RSSI. Ensayos y promedio vs curva teórica.....	50
Figura 36 Ejemplo de funciones prototipo y definiciones CoAP en Contiki OS .....	51
Figura 37 Definición recursos CoAP en lenguaje C .....	51
Figura 38 Mapa del despliegue en instalaciones de ELT .....	52
Figura 39 Imágenes de luminarias instaladas con dispositivo diseñado incorporado ....	54
Figura 40 Capturas de tráfico en despliegue. Filtro por protocolo de aplicación CoAP	55
Figura 41 Análisis de tráfico. Tráfico RPL vs tráfico CoAP.....	56
Figura 42 Mapa de despliegue en instalación con rangos de cobertura adjuntos.....	57
Figura 43 Imágenes sustitución de puntos de luz. Instalación de STELARIA .....	58
Figura 44 Mapa del despliegue en STELARIA. Puntos encendidos enviando tráfico...	59
Figura 45 Frontal CoAP. Filtrado por MAC de nodo. Consulta de recursos .....	59

Figura 46 Ejemplo de perfil de iluminación diseñado. 7 niveles y 45% de ahorro.....	60
Figura 47 Ejemplo de medida en equipo reemplazado.....	61
Figura 48 Topologías de red ofrecidas por el estándar IEEE 802.15.4.....	68
Figura 49 Estándar IEEE 802.15.4. Frecuencias, modulaciones y tasas disponibles.....	68
Figura 50 Trama IEEE 802.15.4. PPDU y MPDU.....	69
Figura 51 Cuadro de máxima potencia radiada permitida. Normativa EN 300 220-1 ...	70
Figura 52 Pila de protocolos de BLE.....	71
Figura 53 Canales frecuenciales disponibles para BLE. Canales de datos y advertising	71
Figura 54 Enlaces disponibles durante ejecución del algoritmo RPL.....	74
Figura 55 Rutas de encaminamiento seleccionado tras ejecución de RPL.....	74
Figura 56 Ejecución de procesos en Contiki OS. Preemptive y cooperative.....	75
Figura 57 Ejemplo de bloque de control e hilo de ejecución para un proceso.....	76
Figura 58 Cabecera Nivel MAC estándar IEEE 802.15.4.....	76
Figura 59 Cabecera IPHC en 6LoWPAN.....	77
Figura 60 Posibles niveles de compresión en dirección IPv6 en 6LoWPAN.....	78
Figura 61 Funcion handler periodic timer. Gestión de las rutas RPL.....	80
Figura 62. Inicialización del algoritmo RPL y procesos Contiki asociados.....	80
Figura 63 Gestión mensajes DIO. Timer asociado y función de control.....	81
Figura 64 Función handler para gestión de envío de mensaje DIO de forma periódica	82
Figura 65 Actualización del timer para envío de mensaje DIO.....	82
Figura 66 Planificación para mensajes DAO. Disparado tras recepción de DIO.....	83
Figura 67 Proceso principal para pila TCP/IP. Inicialización y ejecución del proceso..	84
Figura 68 Función handler del proceso TCP/IP. Ejecutada al recibir trama.....	85
Figura 69 Funciones TCP IP Handler tras recepción de mensaje.....	86
Figura 70 Configuración de los clocks para envío de mensaje UDP como cliente.....	86
Figura 71 DEP. Comunicación GFSK 40 kbps. Portadora 868 MHz.....	90
Figura 72 Curvas BER teórica y práctica para canal simulado.....	90
Figura 73 Curva de potencia recibida vs distancia recorrida hasta 200 m.....	93
Figura 74 Driver ILCPRO 75W de ELT con tecnología eSMART.....	99
Figura 75 Integración de dispositivo STELARIA sobre luminaria.....	99
Figura 76 Módulo de comunicaciones y procesado de datos.....	100
Figura 77 Software de monitorización y gestión. Interfaz Web multi-dispositivo.....	101
Figura 78 Arquitectura software. Módulo de gestión, módulo de interoperabilidad y aplicación de usuario (modelo vista-presentador).....	101
Figura 79 STELARIA. Ejemplo de localización y estado de cada luminaria instalada	103
Figura 80 Estadísticas de un ejemplo de sistema de gestión municipal de padrón.....	104
Figura 81 Herramienta de visualización de tipo de actividad para una localidad.....	105
Figura 82 Sistema de planificación de instalaciones. SLUX.....	105
Figura 83 Gráficas referentes a un punto de luz instalado.....	106

## Lista de Tablas

---

Tabla 1 Comparativa en base a criterios de selección para tecnologías estudiadas .....	26
Tabla 2 Comparativa de prestaciones microcontroladores 32 bits vs 8 bits.....	35
Tabla 3 Direcciones MAC y topología de los dispositivos instalados .....	53
Tabla 4 Direcciones IPv6 de los dispositivos instalados .....	53
Tabla 5 Tabla resumen de la instalación .....	61
Tabla 6 Tabla comparativa de coste de consumo energético entre luminarias .....	62
Tabla 7 Tabla comparativa de coste de consumo incluyendo STELARIA.....	62
Tabla 8 Tabla de ahorro económico para la instalación .....	62
Tabla 9 Procesos intervinientes en Contiki OS para pila TCP/IP .....	87
Tabla 10 Sensibilidad RX vs tasa de transmisión de datos .....	92



MEMORIA



## 1. Introducción

En este primer apartado se realiza una introducción al ámbito de trabajo, explicándose el estado del arte referente a las ciudades inteligentes, el auge de las plataformas de telegestión así como una breve introducción al concepto de alumbrado inteligente. Del mismo modo, se recogen los objetivos principales planteados así como el reparto de tareas llevado a cabo para alcanzar los objetivos mencionados.

### 1.1. Antecedentes

#### 1.1.1 Ciudades y Alumbrado Inteligente

Uno de los objetivos fundamentales de las ciudades inteligentes consiste en alcanzar la interoperabilidad de los servicios municipales. Esto conlleva la capacidad de comunicar información de distintas áreas de gestión como pueden ser la administración digital, la gestión del agua, tratamiento de residuos, movilidad, demografía o alumbrado público entre muchos otros. Dicha integración suele disponer de un alto nivel de complejidad debido a la existencia de estructuras heterogéneas y a la ausencia de estándares tecnológicos. Afortunadamente, las últimas tendencias en el ecosistema IoT (Internet of Things) permite abordar esta disparidad de una manera optimista [1].

En el sector de la iluminación, se está viviendo de forma paralela una doble revolución tecnológica en un intervalo de tiempo muy pequeño. En primer lugar, la irrupción de forma contundente de la tecnología SSL (Solid State Lighting), también denominada tecnología LED, que supone hoy por hoy una realidad extendida [2]. La tecnología SSL se ha convertido en un claro impulsor de la segunda revolución que consiste en dotar conectividad al conjunto de luminarias ya instaladas que conforman la infraestructura de alumbrado público.

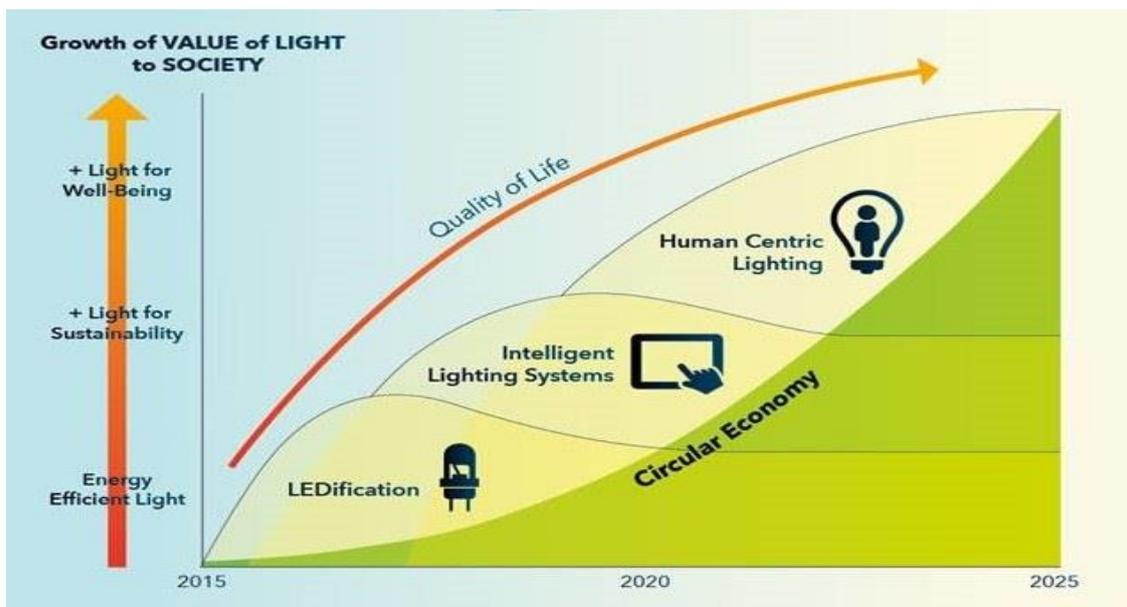


Figura 1 Previsiones de evolución de la tecnología LED

El alumbrado y la iluminación es uno de los servicios públicos que mayor gasto económico acarrear. El pasado año 2017, el gasto económico en concepto de alumbrado público alcanzó la cifra de 955 millones de euros [3]. Las empresas que trabajan en el sector de la iluminación, no solo buscan una transformación hacia la tecnología LED, sino que tratan de dotar de inteligencia a cada punto de luz, con el objetivo de crear una red de dispositivos gestionados de forma remota, cuyo consumo sea únicamente el necesario.

Para que todo esto sea posible, el equipo de control de una luminaria debe ser capaz de gestionar y almacenar información, así como ser capaz de procesar dicha información recibida mediante arquitecturas digitales basadas en microprocesador. Además, en la mayoría de las ocasiones el intercambio de información debe ser bidireccional. Dicha información se debe mostrar al usuario de forma sencilla e intuitiva mediante una aplicación web que permita una representación geolocalizada de las luminarias. En todo momento deberá garantizarse un alto nivel de seguridad, privacidad y fiabilidad del conjunto del sistema, asegurándose de esta forma el servicio de una infraestructura crítica como es el alumbrado público.

La red de alumbrado público se encuentra presente en los lugares más remotos y cubre todos los lugares habitados del territorio, proporcionando desde su privilegiada posición cerca de los ciudadanos y conectada a la red eléctrica no solo el imprescindible servicio de alumbrado, sino también una plataforma ideal para servir de soporte a otros muchos servicios.

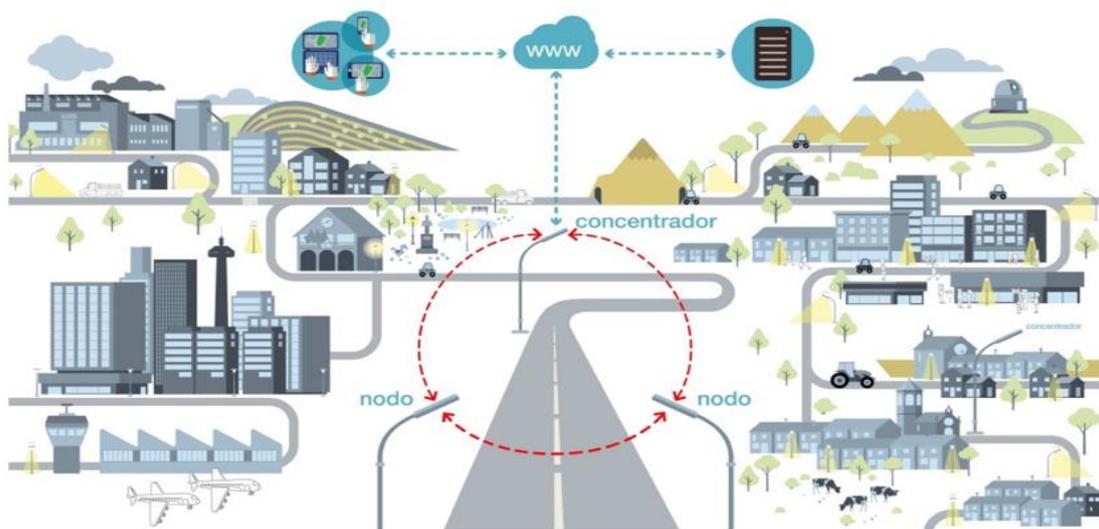


Figura 2 Esquema de red de comunicaciones a través del alumbrado público

Por lo tanto, cada luminaria perteneciente al despliegue de alumbrado público puede convertirse, de facto, en un nodo con capacidad de gestionar datos (data-enabled-lighting) [4]. Más allá de ser un punto de luz, se convierte en un servicio que proporciona conectividad y acceso a Internet formando parte de la infraestructura IoT.

Este acceso a Internet y la capacidad de gestionar cada punto de luz individualmente y de forma remota junto con la transformación hacia la tecnología LED, aporta una mayor

eficiencia lumínica, una iluminación de mayor calidad y un incremento en la vida útil de la luminaria. Además se podrá obtener información acerca del estado de cada punto de luz, facilitando las tareas de mantenimiento. Del mismo modo, se podrá regular el nivel de iluminación de forma individual consiguiendo considerables ahorros energéticos y por lo tanto económicos.

### 1.1.2 Marco General del Proyecto

Este proyecto forma parte del ámbito de las redes de comunicaciones, en especial de las redes de comunicaciones inalámbricas para proporcionar acceso a Internet al sector del alumbrado público. En este trabajo fin de máster se llevará a cabo el diseño, estudio e implementación de una red de comunicaciones mallada e inalámbrica con el fin incorporar conectividad a una de las infraestructuras más grandes como es el alumbrado público exterior.

Dentro de las alternativas que ofrece el mercado, se encuentran conviviendo distintas tecnologías de conectividad como Bluetooth®, WiFi o ZigBee entre muchas otras [5]. Para este proyecto se realizará el diseño de una red de comunicaciones híbrida basada en radiofrecuencia sub-1GHz, haciendo uso de las bandas no comerciales ISM (Industrial, Scientific and Medical) mediante frecuencia portadora a 868 MHz para Europa y 915 MHz para EE.UU. Existirán un conjunto reducido y concreto de dispositivos perteneciente a la red que dispondrán de acceso a Internet haciendo uso de la red de telefonía móvil, empleando las bandas 2G y 3G.

A diferencia de otras topologías de red clásica dentro del mundo IoT y en especial del alumbrado público, el uso de una red híbrida, conlleva una reducción de coste no despreciable pero dificulta el despliegue y la implementación de la red de comunicaciones. Dada la cercanía que existe entre las luminarias que conforman la red de alumbrado público, se puede hacer uso de un radioenlace dentro de banda no licenciada para dispositivos SRD (Short Range Devices). Esto supone un primer requisito de alcance y cobertura en el diseño de la red así como un punto de inflexión de cara a seleccionar la tecnología de comunicación a emplear.

Partiendo de las premisas iniciales, se busca en todo momento hacer uso de estándares y protocolos abiertos que garanticen la interoperabilidad de los dispositivos con otros pertenecientes tanto al sector del alumbrado público como a otros servicios municipales. Toda esta generación y almacenamiento de información, permite de forma posterior acercarse a los conceptos de procesado de datos masivos (Big Data) y aprendizaje automático (Machine Learning) tan aclamados a día de hoy.

## 1.2 Objetivos y Alcance

Los principales objetivos de este trabajo fin de máster son:

- Diseñar una red de comunicaciones basada en RF haciendo uso de estándares y protocolos abiertos.
- Realizar una emulación del diseño para posterior validación de la red.
- Implementar el conjunto de protocolos seleccionados sobre un dispositivo embebido y realizar un diseño electrónico comercial y fabricable.
- Integrar la red de comunicaciones implementada sobre el sistema STELARIA. Realizar el despliegue y puesta en marcha sobre proyecto real de alumbrado público.
- Obtener y analizar los resultados en campo para validar el sistema.

Para conseguir alcanzar los objetivos planteados previamente se han llevado a cabo una serie de tareas que se describen a continuación:

En primer lugar se ha realizado un estudio en lo que a la tecnología de comunicaciones se refiere. Se han analizado las alternativas ya existentes en el mercado, destacando los pros y desventajas de cada alternativa. Se ha realizado una selección de aquellas más óptimas de cara a un despliegue de dispositivos orientados al sector del alumbrado público y las ciudades inteligentes.

Una vez escogida la tecnología, se ha realizado un análisis de dispositivos hardware que permiten dicha implementación de cara a seleccionar una serie de componentes capaces de implementar la capa física por la que se ha optado. Posteriormente, se han estudiado y analizado los estándares y protocolos desde el nivel de aplicación hasta el nivel físico, basados en redes de sensores, teniendo en cuenta siempre las limitaciones en referencia a las capacidades de cómputo y de memoria del dispositivo seleccionado.

Después de obtener una primera versión del firmware y una implementación sobre dispositivo, se ha procedido a emular una red mediante software dedicado para este propósito. Tras validar el diseño de la red y la funcionalidad de los dispositivos, se ha realizado un diseño electrónico y se ha iniciado un proceso de fabricación y puesta en marcha de producción de cara a validar la comercialización del sistema.

Finalmente, se ha integrado la red diseñada e implementada sobre un sistema real como STELARIA. En concreto, el despliegue de red ha tenido su puesta en marcha para un proyecto de renovación de alumbrado público cuyo pliego de condiciones recogía la necesidad de integrar una red mallada como la diseñada que permitiese realizar una gestión remota y eficiente del alumbrado.

Por último se han analizado los resultados obtenidos en referencia a las estimaciones de eficiencia, ahorro y mejoras que recogían las premisas iniciales de este proyecto.

1.2.1 Cronograma

En este apartado se recoge en formato tabla el cronograma del presente trabajo fin de máster con el objetivo de resumir el reparto temporal de tareas que permite cumplir los objetivos mencionados en el apartado previo:

	2017				2018							
	09	10	11	12	01	02	03	04	05	06	07	08
Estudio de las tecnologías de comunicaciones para redes de sensores												
Estudio y selección de protocolos												
Emulación en entorno software Validación de la implementación												
Diseño electrónico												
Proceso de producción fabricación y test Pruebas funcionales												
Despliegue de red en proyecto real de alumbrado público												
Análisis de resultados												
Redacción de la memoria												

1.3 Estructura de la Memoria

Este trabajo fin de máster queda estructurado de la siguiente forma:

En el capítulo 2 se realiza un estudio de la tecnología y de los protocolos más empleados en las redes de comunicaciones. Desde una selección tecnológica que proporcione soporte físico hasta el nivel de aplicación, en este apartado se describen y seleccionan los distintos protocolos que conformarán la red de comunicaciones implementada.

En el capítulo 3 de esta memoria, se analiza la implementación llevada a cabo con el objetivo de obtener un dispositivo funcional. Se argumenta la necesidad de utilizar un sistema operativo en tiempo real y se introduce el sistema operativo Contiki OS. Del mismo modo, se recoge el estudio y selección de la familia del microcontrolador empleado que dará soporte al sistema operativo y al resto de requisitos iniciales planteados previamente a la selección del dispositivo. Se explica la emulación del dispositivo que se ha realizado y la validación del funcionamiento sobre cada nodo del conjunto protocolos seleccionados en el capítulo previo.

En referencia al desarrollo del propio dispositivo, se recoge el diseño electrónico llevado a cabo de un total de 3 placas de circuito impreso. Como una de las partes más importantes de un producto, se explicarán los procesos de fabricación y test de la electrónica, así como la interacción de dichos procesos con un ERP (Enterprise Resource Planning). Finalmente

se presenta la validación de la funcionalidad de los dispositivos y su correcto funcionamiento mediante un pequeño despliegue de nodos en una instalación.

En el capítulo 4, se estudia un caso de implementación real del sistema desarrollado. Tras superar un proceso de fabricación y validación del sistema en un entorno de producción real, se procede a realizar un despliegue de red utilizando el diseño recogido en este proyecto. Se valida el sistema y se realiza un estudio de los resultados cosechados en un caso de aplicación real.

Por último, en el capítulo 5 se recogen una serie de conclusiones que permiten verificar que el diseño realizado se adapta a las premisas y requisitos iniciales. Se verifica del mismo modo que la implementación llevada a cabo es válida y que tras el despliegue de dispositivos en un caso de proyecto de iluminación, el sistema cumple las funcionalidades previstas inicialmente.

Adjunto al proyecto quedarán las referencias bibliográficas así como una serie de anexos donde se puede encontrar información ampliada de los puntos principales del trabajo.

## 2. Selección de Protocolos

En los últimos años se ha registrado un auge de gran importancia en el mundo de las TIC (Tecnologías de Información y Comunicación), haciendo especial énfasis en el desarrollo y despliegue de redes de sensores y dispositivos inalámbricos. También conocidas como WSN (Wireless Sensor Networks), dan un paso más allá tratando de ofrecer una conexión con la red más importante a nivel global, Internet.

Los mercados plantean una tendencia a disminuir el coste del dispositivo manteniendo los conceptos de escalabilidad e implementación sencilla. Esto se consigue a través de un coste de memoria de almacenamiento muy pequeño y un coste energético ínfimo, donde en muchas ocasiones no se dispondrá de alimentación para el sistema [6]. El desarrollo y despliegue de las redes de comunicaciones para dispositivos IoT y redes WSN conllevan un reto a día de hoy por una clara falta de estandarización. Por ello, el objetivo principal de este proyecto es desarrollar una red basada en estándares y protocolos abiertos.

### 2.1. Nivel Físico y Nivel de Enlace. Selección de la Tecnología

En este primer punto se analizan alguna de las tecnologías de comunicación inalámbricas que existen en el mercado y se emplean en el ámbito de las WSN. Posteriormente, será necesario definir un conjunto de criterios de cara a tomar una decisión tecnológica que supondrá el soporte de toda la pila de protocolos que se decida implementar.

#### 2.1.1. Tecnologías Presentes en el Mercado

##### Bluetooth Low Energy

El Bluetooth de baja energía (Bluetooth Low Energy o BLE), es un subconjunto del estándar Bluetooth v4.0. Dispone de una pila de protocolos (ver Anexo B) en referencia a la capa OSI completamente nueva y orientada a conexiones sencillas en aplicaciones de muy baja potencia (dispositivos dependientes de batería o pila).

Ofrece una tasa de transmisión de datos desde 125 kbps hasta 2 Mbps. Una de sus mayores ventajas reside en la aceptación obtenida por parte de las grandes plataformas a día de hoy como IOs, Android, Microsoft o Linux entre otras y su compatibilidad con éstas.

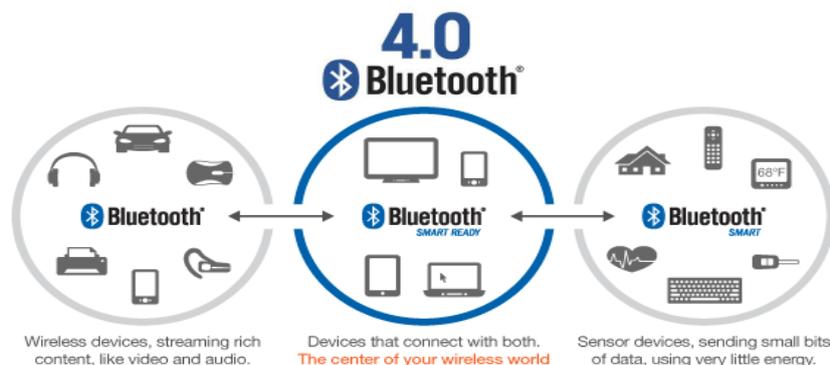


Figura 3 Bluetooth 4.0. Bluetooth Smart o Bluetooth Low Energy

### Estándar IEEE 802.11. Wi-Fi

Wifi es una tecnología de interconexión inalámbrica para dispositivos electrónicos. Wi-Fi es la marca de la alianza Wi-Fi, cuya organización de carácter comercial aprueba y certifica los equipos que implementan el estándar IEEE 802.11, relacionado con redes inalámbricas de área local. Al igual que Bluetooth, trabaja de forma habitual sobre la banda ISM de 2.4 GHz con posibilidad de operar en la banda de 5 GHz.

Se trata de un estándar pensado para soportar las comunicaciones de las redes tradicionales de una forma inalámbrica, con un gran ancho de banda disponible y con un MTU (Maximum Transmission Unit) grande donde no existen unos requerimientos de batería elevados.



Figura 4 Estándar 802.11. Wi-Fi

### SigFox

SigFox es una red alternativa para dispositivos IoT con origen francés, que dispone de un amplio despliegue actualmente, especialmente a nivel europeo. Utiliza su propio despliegue de red e infraestructura de comunicaciones como alternativa a otras soluciones que emplean la red de telefonía móvil [7]. La tasa de transmisión de datos de SigFox es de 100 bps hasta un máximo de 600 bps.

Emplea la banda libre para dispositivos SRD en 868 MHz en Europa y 915MHz en EE.UU. Como contrapartida, se trata de una solución totalmente cerrada dado que para poder comunicarte a través de la red, necesitas comprar la implementación y la electrónica disponibles a través de un único canal de distribución, la propia empresa SigFox. La interoperabilidad con otras plataformas se realiza mediante API (Application Programming Interface) en nube, consumiendo recursos de sus propios servidores.

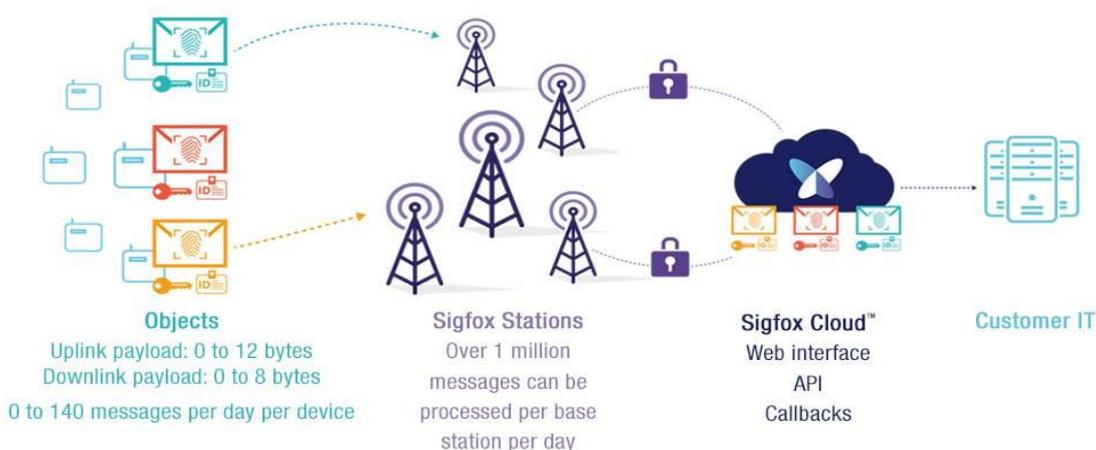


Figura 5 Esquema de red SIGFOX. Solución cerrada con acceso mediante API

### LoRa y LoRaWAN

LoRa es un tipo de modulación chirp patentada por la marca Semtech que muestra una alta tolerancia ante interferencias y alta sensibilidad en recepción de modo que empleando las bandas no licenciadas de comunicaciones (868 MHz en Europa y 915 MHz en EE.UU) logra alcances de hasta 20 km. La tasa de transmisión de datos de LoRa es 250 bps.

Sobre el soporte físico establecido por LoRa, se implementa LoRaWAN [8], una capa de enlace para dispositivos IoT con bajo overhead. Está orientado a bajas transferencias de datos y bajo consumo energético de los dispositivos. Implementa una red en estrella, donde los dispositivos se comunican directamente con el dispositivo concentrador que proporciona acceso a Internet.

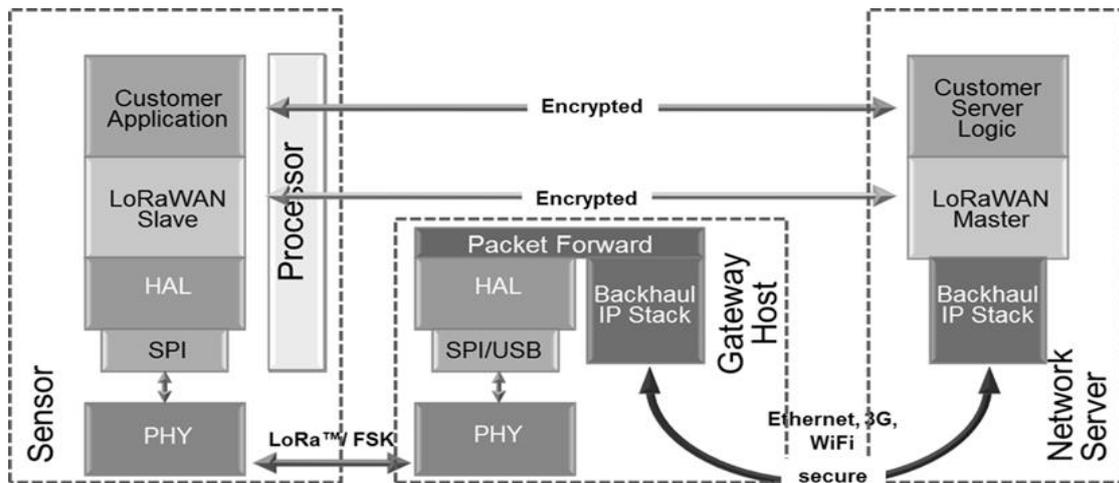


Figura 6 Dispositivo LoRaWAN. Conexión con Servidor

### Estándar IEEE 802.15.4

Se trata de un estándar que define el nivel físico y el nivel de enlace para redes inalámbricas con baja tasa de transmisión de datos. Da soporte a distintas especificaciones como ZigBee, Thread o 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks) que ofrecen una solución completa mediante la implementación de los niveles superior en la pila de protocolos que este estándar no cubre.

Su implementación puede llevarse a cabo sobre los canales físicos de 868 MHz (Europa), 915 MHz (EE.UU) y 2,4 GHz. Permite una tasa de transmisión hasta 250 kbps, muy superior otras tecnologías como SigFox o LoRaWAN descritas anteriormente [9].

ZigBee	6LoWPAN	Thread	ISA100	Wireless HART
ZigBee protocol	Protocol	Protocol (e.g. CoAP)	ISA protocol	HART protocol
ZigBee transport	UDP	UDP/TCP	UDP	HART TCP like
ZigBee networking	6LoWPAN	6LoWPAN, DTLS, distance vector routing	6LoWPAN	HART addressing/routing
IEEE 802.15.4 MAC	IEEE 802.15.4 MAC	IEEE 802.15.4 MAC	Upper data link ISA100 IEEE 802.15.4 MAC	HART TDMA - hopping
IEEE 802.15.4 2.4 GHz • O-QPSK	IEEE 802.15.4 2.4 GHz • O-QPSK	IEEE 802.15.4 2.4 GHz • O-QPSK	IEEE 802.15.4 2.4 GHz • O-QPSK	IEEE 802.15.4 2.4 GHz • O-QPSK

Figura 7 Distintas implementaciones haciendo uso de estándar IEEE 802.15.4

2.1.2. Criterios de Selección y Comparativa

De cara a realizar una selección tecnológica, previamente se han de recoger parámetros de valoración. En este caso se han utilizado como criterios de selección los siguientes parámetros:

- Tasa de datos y ancho de banda
- Alcance
- Frecuencia portadora y dimensiones
- Consumo energético y vida útil del dispositivo

En referencia a la tasa de datos se deben tener en cuenta las siguientes premisas. El dispositivo debe ser capaz de mantener información de estado de la red y realizar el envío de mensajes correspondientes a la capa de aplicación. Del mismo modo, al tratarse de un diseño de red mallada, cada nodo o dispositivo tiene que tener capacidad de reenviar tráfico que no genera él de forma directa. Se estipula que la tasa de transmisión de datos para el correcto funcionamiento del sistema debe situarse entre los 20 kbps y los 100 kbps. Por lo tanto, esto descarta el uso de tecnologías como SigFox o LoRaWAN, dado que las tasas de transmisión de datos ofrecidas son muy bajas (<600 bps).

El alcance supone otro de los parámetros fundamentales de cara a realizar un diseño de dispositivo para una red inalámbrica. La movilidad no será un delimitante, pero sí la distancia entre dispositivos. A modo de ejemplo, la distancia entre puntos de luz de la red de alumbrado público suele situarse en un intervalo de 20-30 metros aproximadamente, aumentando dichas distancias en entornos rurales. Del mismo modo, habrá que disponer de capacidad para combatir la interferencia multicamino, y la existencia de radioenlaces NLOS (Non Line of Sight). Esto descarta el uso de tecnologías como Wi-Fi o Bluetooth dado que en muchas ocasiones el radio de cobertura ofrecido no será suficiente.

La frecuencia portadora de la comunicación suele ser un condicionante a la hora de seleccionar el tamaño del dispositivo o realizar un diseño electrónico. Las dimensiones serán válidas para cualquiera de las tecnologías (portadoras de centenas de MHz) lo que da lugar a tamaños aceptables con capacidad de instalación de una forma sencilla.

Otro de los puntos fundamentales a la hora de escoger un dispositivo para una red WSN suele ser la vida útil del dispositivo, limitada por el consumo energético. Aunque se considera un parámetro de decisión, inicialmente el dispositivo dispondrá de fuente de alimentación externa y por lo tanto no será el criterio principal para nuestra selección.

Propiedad	BLE	802.11	SigFox	LoRa-WAN	802.15.4
Tasa de Transmisión	Correcto	Correcto	Insuficiente	Insuficiente	Correcto
Alcance	Insuficiente	Insuficiente	Correcto	Correcto	Correcto
Dimensiones	Correcto	Correcto	Correcto	Correcto	Correcto
Consumo Energético	>10 años	Insuficiente	>10 años	>10 años	>10 años

Tabla 1 Comparativa en base a criterios de selección para tecnologías estudiadas

Por ello, utilizando como referencia la Tabla 1 se puede justificar que la selección del soporte físico de los dispositivos que conformarán la red será el estándar IEEE 802.15.4. En el Anexo A de esta memoria se encuentra más información referente a este estándar.

## 2.2. Nivel de Red.

### 2.2.1. Protocolo de Red. IPv6 sobre 6LoWPAN

6LoWPAN [10] es un estándar del IETF (Internet Engineering Task Force), RFC 6282, que permite la comunicación mediante el protocolo de red IPv6 y los niveles de aplicación que se basan en éste [11]. Surge ante la necesidad de proporcionar acceso a Internet a multitud de dispositivos, pensando en todo momento en un bajo coste así como unos requerimientos de almacenamiento y consumo energético muy reducidos.

Para ello, los protocolos orientados a redes cableadas o redes inalámbricas dejan de ser una solución factible. Esto es así dado que los protocolos o estándares como IEEE 802.15.4 utilizado en este diseño, ofrecen tamaños de trama muy pequeños (127 bytes en este caso) frente a las redes de acceso a Internet más clásicas como Ethernet, que ofrece un tamaño de trama de 1500 bytes.

Este estándar, se define como una capa de adaptación entre el nivel de enlace y el nivel de red IPv6. El éxito de esta capa de adaptación reside en utilizar un estándar comúnmente utilizado como IPv6. 6LoWPAN realiza la compresión o supresión de campos de la cabecera IPv6 que para una red WSN, donde los niveles de procesado y almacenamiento son limitados, pueden ser irrelevantes. En la Figura 8 se muestra un ejemplo de red 6LoWPAN.

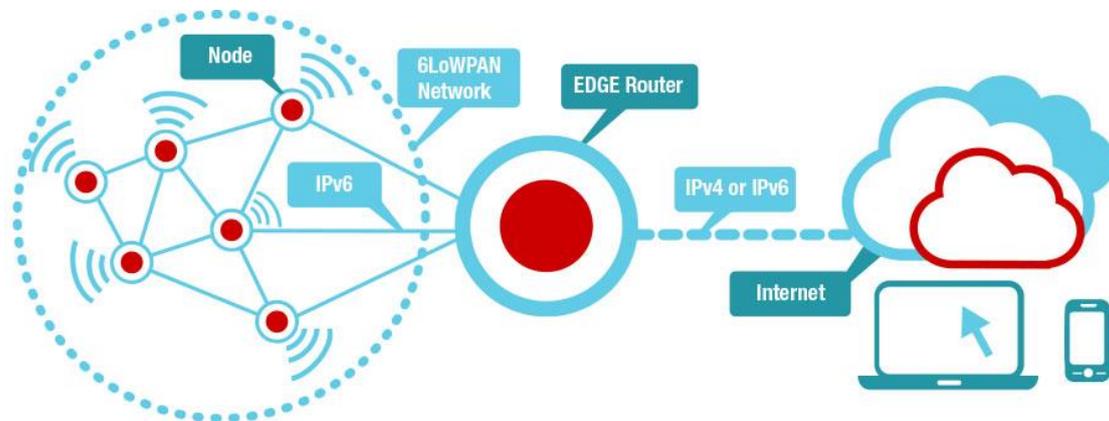


Figura 8 Ejemplo de red 6LoWPAN

Dada la falta de estandarización que existe en el mercado para este tipo de dispositivos, se busca no tener que realizar el pago de cánones o royalties por disponer o hacer uso de tecnología perteneciente a alianzas de empresas o entidades privadas. Este es uno de los motivos por los que una alternativa basada en conectividad IPv6 sobre un estándar de adaptación como 6LoWPAN, supone una solución abierta y de plenas garantías tanto de escalabilidad como de interoperabilidad libre de royalties.

En la Figura 9 se recoge una imagen ilustrativa de la evolución del mercado donde se recoge la tendencia mencionada hacia un estándar abierto.

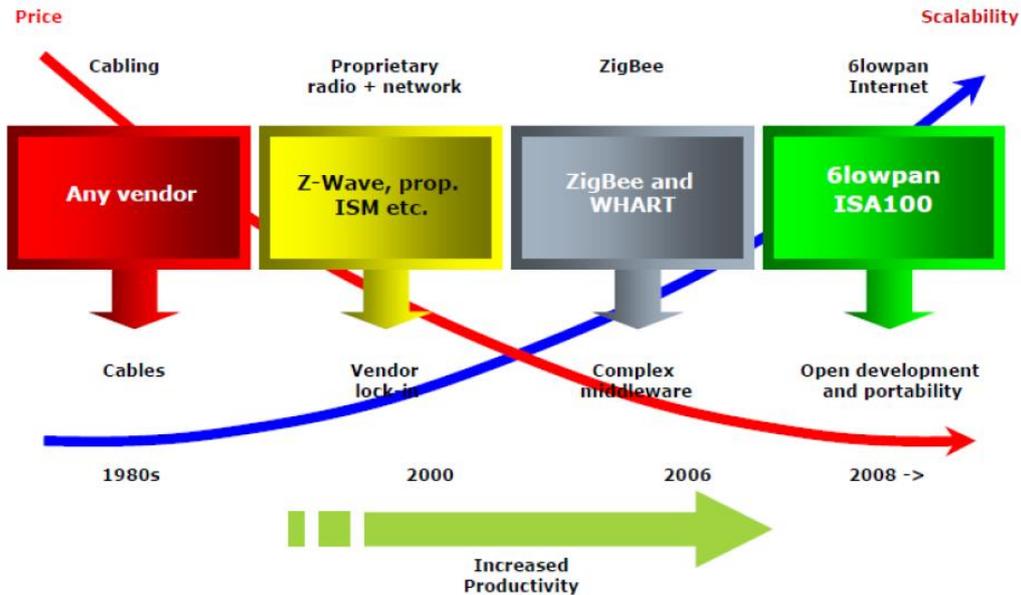


Figura 9 Comparativa precio vs escalabilidad. Tendencia del mercado

### 2.2.2. Protocolo de Enrutamiento. RPL

En las redes inalámbricas y más en especial en las WSN, se asume que los enlaces serán no fiables, asimétricos y con topología variante. Por ello, el encaminamiento se convierte en un apartado crítico.

Los algoritmos como RIP (Routing Information Protocol), RFC 2453 [12] u OSPF (Open Shortest Path First), RFC 5340 [13] empleados en las redes de Internet tradicionales no son soluciones adecuadas. Éstos no encajan en el ámbito de las redes inalámbricas dado que no están concebidos para enlaces asimétricos o topologías dinámicas que utilizan actualizaciones periódicas y requieren un volcado de información que supone un incremento nada despreciable en términos de consumo de potencia y ancho de banda.

Con el objetivo de dar solución a la problemática planteada anteriormente, se desarrolla RPL (Routing over low Power Lossy Networks), RFC 6550. Se trata de un algoritmo de encaminamiento proactivo (determina las rutas independientemente del patrón de tráfico) e independiente de la tecnología subyacente que ofrece un estándar abierto para el proceso de encaminamiento orientado a redes IP.

El algoritmo RPL [14] está basado en los conceptos de métrica, función objetivo (OF) y rango. La métrica, indica cuál es el factor cuantitativo que se utilizará para determinar el enlace por el cual se realizará la retransmisión de un mensaje. RPL permite configurar las métricas de encaminamiento para una red de dispositivos en base a factores como el número de saltos, la latencia del enlace o el ETX (Expected Transmission Count). La función objetivo específica como RPL utiliza o combina la monitorización de la métrica seleccionada para obtener el rango, un valor numérico que representa la lejanía hasta el nodo concentrador con acceso a Internet. El algoritmo mencionado permite conformar una red de tipo árbol desde los nodos sensores hacia el nodo concentrador. En la Figura

10 se muestra dicha estructura. En el Anexo C de esta memoria se recoge un ejemplo de aplicación del algoritmo RPL.

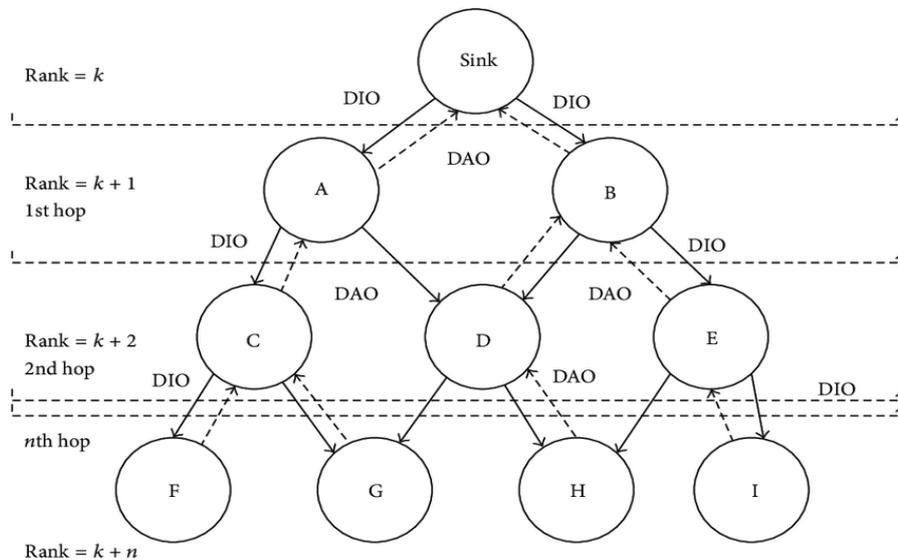


Figura 10 DODAG, Destination Oriented Directed Acyclic Graph

En resumen, el algoritmo RPL proporciona un encaminamiento en base a un estándar abierto, basado en un código y una aritmética compacta y más sencilla que los algoritmos de encaminamiento para las redes cableadas.

### 2.3. Nivel de Transporte. UDP

El nivel de transporte se corresponde con el tercer nivel del modelo TCP/IP. Ofrece un servicio de comunicación extremo a extremo. Utiliza los servicios del nivel de red para el envío de paquetes y dentro de sus funcionalidades destacan la identificación de aplicación o proceso mediante número de puerto, soporte de múltiples aplicaciones sobre un único host o, en ocasiones, proporcionar garantías de entrega de mensaje.

En las WSN se encuentra ampliamente extendido el uso del protocolo UDP. Es un protocolo de transporte no orientado a conexión que no garantiza la entrega de los mensajes. Por el lado contrario, no requiere establecer una conexión previa entre cliente y servidor para realizar una comunicación, su overhead es muy reducido (cabecera de 8 bytes) y aporta identificación del proceso mediante identificador de puerto.

En la Figura 11 se recoge una comparativa entre las cabeceras de UDP y otro de los protocolos de transporte más extendidos en las redes de comunicaciones, TCP. En las redes de comunicación tradicionales y cableadas se opta por UDP cuando existen requisitos de overhead o retardo muy estrictos y el uso de TCP no es factible, o en protocolos de aplicación muy sencillos como DNS (Domain Name System).

A modo de ejemplo y tomando como referencia la Figura 11, la cabecera para un fragmento TCP puede variar entre 20 y 60 bytes. El estándar IEEE 802.15.4 dispone de un MPDU (MAC Protocol Data Unit) de 127 bytes, por lo que utilizar TCP supondría

entre un 15% y un 55% del espacio disponible en la trama. Además, en base a la selección del nivel de red donde se contempla la capa de adaptación 6LoWPAN, existe la posibilidad de comprimir la cabecera UDP hasta 4 bytes suponiendo el uso de tan solo un 3% del MPDU disponible para proporcionar un nivel de transporte.

TCP Segment Header Format								
Bit #	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Sequence Number							
64	Acknowledgment Number							
96	Data Offset	Res	Flags			Window Size		
128	Header and Data Checksum				Urgent Pointer			
160...	Options							

UDP Datagram Header Format								
Bit #	0	7	8	15	16	23	24	31
0	Source Port				Destination Port			
32	Length				Header and Data Checksum			

Figura 11 Comparativa cabeceras TCP y UDP

#### 2.4. Nivel de Aplicación. CoAP

En referencia al nivel de aplicación, al igual que se ha ido exponiendo la problemática existentes en las redes WSN de ancho de banda, capacidad de procesado, almacenamiento y consumo, los protocolos empleados en las redes de comunicación tradicionales como HTTP (Hypertext Transfer Protocol) o FTP (Hypertext Transfer Protocol) resultan francamente complejos de implementar.

Es importante resaltar que el objetivo de un dispositivo perteneciente a una WSN no suele ser realizar la descarga de una página web o un conjunto de ficheros. El objetivo fundamental de este tipo de redes es monitorizar un conjunto de variables y actualizar dicho valor sobre una base de datos que reside en un servidor direccionado o descargar información limitada que permita actuar sobre el propio dispositivo.

El protocolo CoAP (Constrained Application Protocol), RFC 7252 [15], surge ante la necesidad de disponer de un protocolo estándar y de desarrollo abierto cuyos requerimientos de ancho de banda y procesado fueran francamente reducidos. El objetivo de dicho protocolo es mantener los servicios basados en la web. Es decir, mantener los conceptos de información gestionados por servidores, recursos identificados por URLs y acceso asíncrono basado en arquitectura REST (Representational State Transfer).

En la Figura 12 se muestra un ejemplo de intercambio de mensajes entre cliente y servidor CoAP. El servidor expone un conjunto de recursos (en este caso temperatura y nivel de luz). El cliente solicita el conjunto de recursos (identificados mediante URLs) disponibles y posteriormente solicita el valor del recurso temperatura.

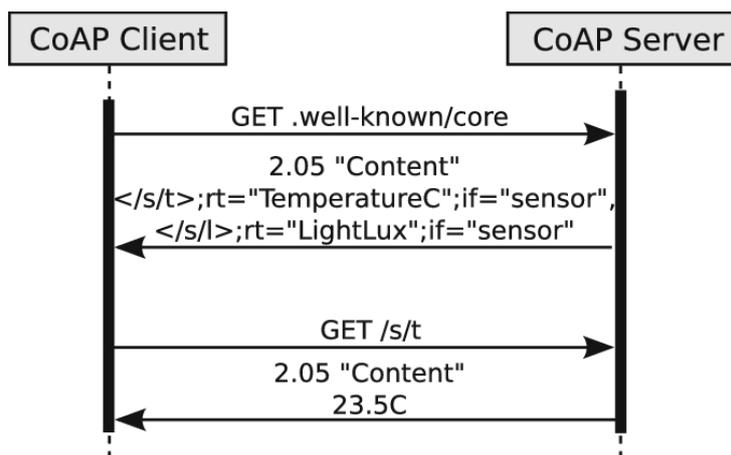


Figura 12 Intercambio mensajes entre cliente y servidor CoAP

Entre sus características más destacables cabe resaltar el uso de una cabecera muy sencilla de tan solo 4 bytes (ver Figura 13) y el uso del protocolo UDP como nivel de transporte.

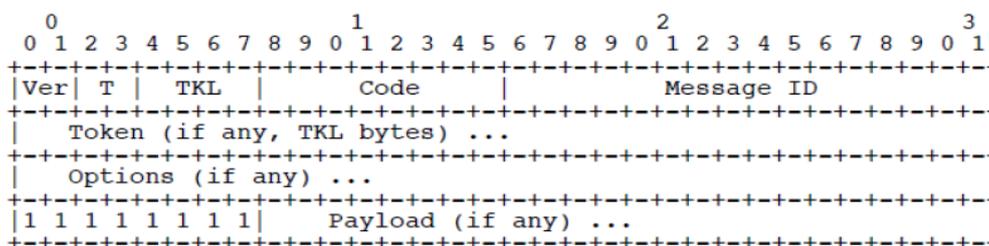


Figura 13 Cabecera CoAP

De esta forma quedarían definidos cada uno de los niveles que conformarán la red de comunicaciones, cuyo diseño e implementación conforman el objetivo principal de este proyecto. La Figura 14 recoge a modo de esquema una representación gráfica de la pila de protocolos empleada.

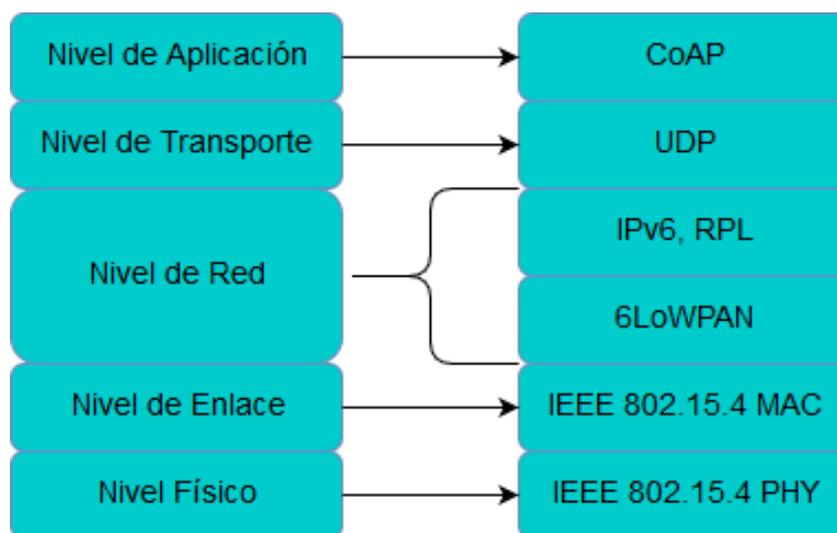


Figura 14 Pila de protocolos obtenida para diseño de red de comunicaciones



### 3. Implementación de la Red de Comunicaciones

---

Una vez seleccionado el conjunto de protocolos a emplear, el siguiente paso consiste en realizar una implementación física y real de los dispositivos. En capítulos previos de esta memoria se ha analizado la necesidad de disponer de un dispositivo de bajo coste, basado en los principios de ancho de banda reducido, requisitos de almacenamiento y procesado restringidos y bajo consumo energético. Esto conlleva trabajo tanto en un ámbito software como de diseño electrónico embebido.

En este capítulo se recogen las pautas llevadas a cabo para la implementación de los dispositivos que conformarán la red de comunicaciones. Será necesario disponer, dada la complejidad y la cantidad de procesos intervinientes, de un sistema operativo de tamaño reducido, capaz de gestionar las distintas tareas asociadas a la red de comunicaciones y al propio dispositivo como las interrupciones. Posteriormente, se llevará a cabo una emulación de los dispositivos que permita validar la selección del S.O. y la implementación de la pila de protocolos seleccionada.

En base a los requisitos técnicos, como memoria de almacenamiento o capacidad de procesado, será necesario escoger un dispositivo embebido sobre el que compilar el sistema operativo. Una vez evaluada la implementación software de los nodos, será necesario realizar un diseño electrónico basado en PCB (Printed Circuit Board), que permita generar un proceso de fabricación y replicación de los dispositivos diseñados.

Se llevarán a cabo procesos de fabricación y test, así como un conjunto de ensayos y pruebas que permitan validar la comunicación a través del espacio radioeléctrico. Finalmente, se realizará un despliegue para la validación completa del sistema.

#### 3.1. Tipo de Dispositivos

En primer lugar, se realiza una diferenciación entre los tipos de dispositivo que conformarán la red. Las dos versiones de nodo serán:

- **Nodo simple:** encargados de monitorizar las variables referentes a los sensores, dispositivos o actuadores que finalmente se implementen en el diseño. Dispondrán de un interfaz radio que permita realizar la comunicación entre otros nodos simples y nodos concentradores con el fin de realizar el intercambio de información.
- **Nodos concentradores:** serán los encargados de reportar toda la información recibida de los nodos simples hacia el servidor. Dispondrán de un interfaz radio para recibir los mensajes de los nodos simples y además dispondrán de un enlace a Internet para realizar la subida de mensajes hacia el servidor. En este caso, el enlace de conexión a Internet será a través de la red de telefonía móvil mediante acceso basado en tecnología 2G y 3G.

En la Figura 15 se muestra una imagen que representa el funcionamiento de la red.

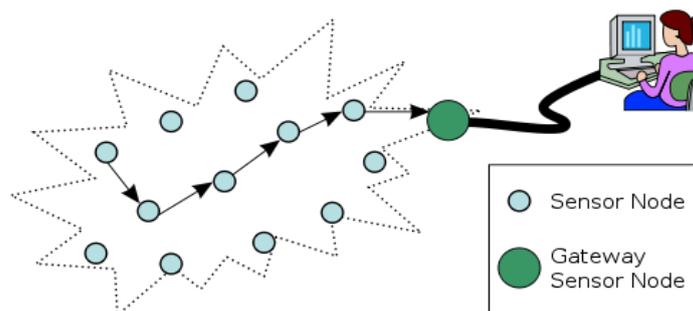


Figura 15 Ejemplo de red WSN

### 3.2. Sistema Operativo. Contiki OS

Contiki OS es un sistema operativo RTOS (Real-Time Operating System), programado en código C y desarrollado para el uso de WSN, basado en arquitecturas electrónicas orientadas a microcontroladores [16]. Dispone de una pila que proporciona soporte TCP/IP y provee de procesamiento multitarea, permitiendo la ejecución de varios procesos sobre un único núcleo de procesamiento, como es el caso de un microcontrolador.

Una de las propiedades que define este sistema operativo es su bajo coste de almacenamiento. Consta de 2 kB de RAM (Random Access Memory) y 40 kB de memoria flash para una configuración estándar. Todos los programas de Contiki OS son procesos disparados por la sucesión de uno o varios eventos. Es decir, son piezas de un código que se ejecuta de forma regular. En el Anexo D de esta memoria se encuentra más información obtenida tras el estudio, análisis y modificaciones de Contiki OS.

Los bajos requisitos de memoria definen Contiki OS como un sistema operativo de bajo coste de almacenamiento. Otro de las ventajas que ofrece es el amplio soporte a nivel de plataforma, es decir, la capacidad de ser implementado en distintas alternativas hardware basadas en microcontrolador o SoC (System on Chip). En la Figura 16 se recoge una tabla que muestra alguna de las plataformas disponibles.

MCU/SoC	Radio	Platforms
TI CC2538	Integrated / CC1200	<a href="#">RE-Mote</a>
nRF52832	Integrated	<a href="#">nRF52 DK</a>
RL78	ADF7023	<a href="#">EVAL-ADF7023DB1</a>
TI CC2538	Integrated	<a href="#">cc2538dk</a>
TI MSP430x	TI CC2420	<a href="#">exp5438, z1</a>
TI MSP430x	TI CC2520	<a href="#">wismote</a>
Atmel AVR	Atmel RF230	<a href="#">avr-raven</a> , <a href="#">avr-rcb</a> , <a href="#">avr-zigbit</a> , <a href="#">iris</a>
Atmel AVR	TI CC2420	<a href="#">micaz</a>
Freescale MC1322x	Integrated	<a href="#">redbee-dev</a> , <a href="#">redbee-econotag</a>
TI MSP430	TI CC2420	<a href="#">sky</a>
TI MSP430	TI CC1020	<a href="#">msb430</a>
TI MSP430	RFM TR1001	<a href="#">esb</a>
Atmel Atmega128 RFA1	Integrated	<a href="#">avr-atmega128fa</a>

Figura 16 Ejemplos de plataformas hardware que soportan Contiki OS

### 3.3. Selección de Dispositivo Embebido

Después de analizar el sistema operativo que dará soporte a la implementación, es necesario encontrar un dispositivo físico que proporcione las prestaciones necesarias para llevar a cabo el desarrollo. Como se ha mencionado previamente, el sistema dispondrá de una implementación electrónica basada en microcontrolador. Antes de seleccionar el dispositivo, es necesario analizar las tareas y los periféricos que se van a requerir.

#### 3.3.1. Requisitos

En primer lugar, será imprescindible disponer de un puerto de comunicaciones para poder realizar la comunicación con el transceptor. Los interfaces más empleados habitualmente para este tipo de comunicaciones entre microcontrolador y periférico externo son UART (Universal Asynchronous Receiver-Transmitter) I<sup>2</sup>C (Inter-Integrated Circuit) o SPI (Serial Peripheral Interface).

Por otro lado, el dispositivo debe incluir puertos asociados a la comunicación con sensores u otros dispositivos. Los interfaces de comunicación pueden ser analógicos o digitales. En caso de ser analógicos, será necesario disponer de un periférico ADC (Analog-Digital Converter) y un DAC (Digital-Analog Converter).

Como requisitos de memoria, en base al sistema operativo Contiki OS, será necesario disponer de al menos 2 kB de memoria RAM y 40 kB de memoria flash. En cuanto al nivel de procesado, uno de los puntos críticos será el ancho de palabra. Es decir, el número de bits que dispondrán las posiciones de memoria del microcontrolador. Los tamaños más utilizados son 8, 32 y 64 bits. El uso de un ancho de palabra u otro suele venir delimitado por la aplicación y uso al que se va a destinar el microcontrolador.

#### 3.3.2. Selección de Arquitectura y Familia de Microcontrolador

En la Tabla 2 se recoge una comparativa con las prestaciones existentes en el mercado referente a las soluciones de 8 y 32 bits más ampliamente utilizadas en el mercado.

Propiedad	32 bits	8 bits
<b>Velocidad Procesador</b>	32-180 MHz	20 MHz
<b>Coste</b>	1,5€	0,5€
<b>Lenguaje Programación</b>	C	Ensamblador y C
<b>Consumo Energético en Bajo Consumo</b>	1500nA	100 nA
<b>Periféricos de Comunicación</b>	3xUART, 2xSPI, I2C, BLE, USB, I2S ...	UART, SPI, I2C
<b>Memorias</b>	1MB Flash, 256KB RAM	32KB Flash, 4KB RAM
<b>Diversidad</b>	Aplicaciones Específicas (Industrial, Inalámbrico)	Limitado

Tabla 2 Comparativa de prestaciones microcontroladores 32 bits vs 8 bits

Una arquitectura basada en ancho de palabra de 32 bits permite disponer de mayores prestaciones a costa, de forma habitual, de un precio más elevado. Más en concreto, los requisitos de memoria y periféricos descritos previamente entran dentro de las prestaciones que ofrece un microcontrolador 32 bits, mientras que no podríamos garantizar esto mismo para un microcontrolador de 8 bits.

El siguiente paso es seleccionar la arquitectura del microcontrolador. En el ámbito de los microcontroladores existen diversas familias como PIC, AVR, x86 o ARM. Cada una de ellas está orientada a un tamaño de palabra específico. Para el diseño de este proyecto se utilizará una arquitectura ARM.

ARM es una arquitectura RISC (Reduced Instruction Set Computer) de 32 bits, el conjunto de instrucciones para este tamaño de palabra más ampliamente utilizado. Desde 2009, los procesadores ARM conforman el 90% de los procesadores RISC de 32 bits embebidos empleados en electrónica de consumo como tablets o telefonía móvil [17].

Dentro de la arquitectura de microcontroladores ARM existen distintas familias en función de la aplicación. En la Figura 17 se recoge un gráfico ilustrativo de las familias mencionadas.

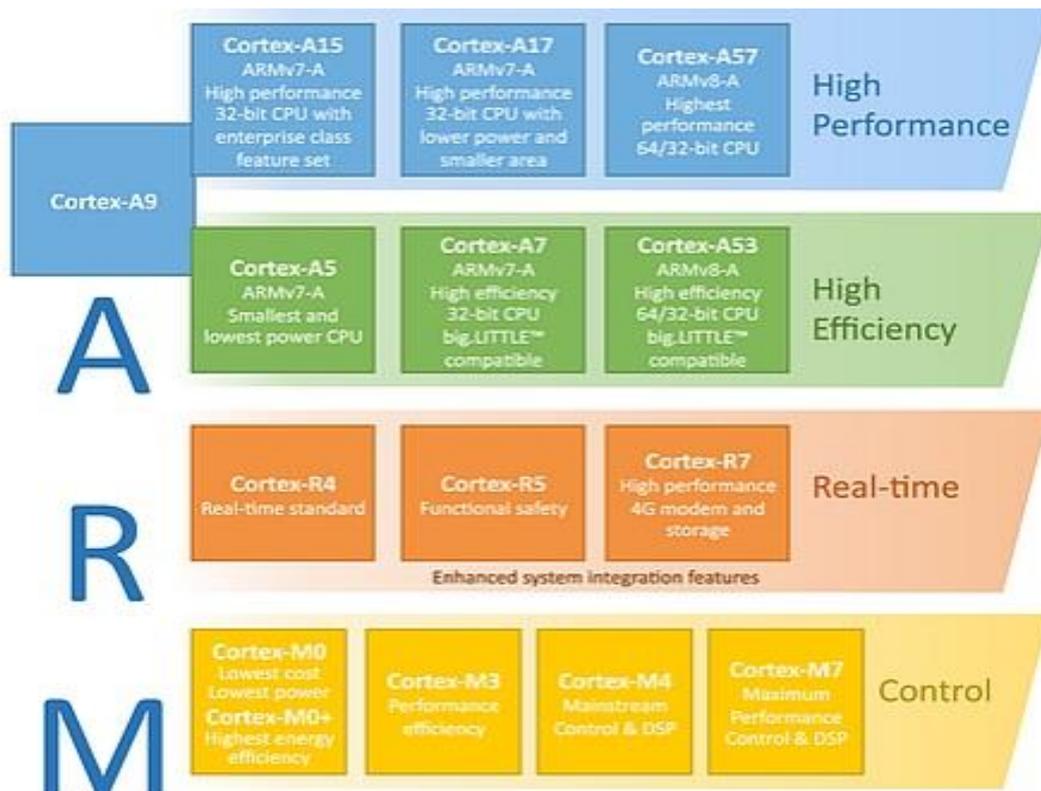


Figura 17 Familias de dispositivos para arquitectura ARM

Se utilizará la serie de microcontroladores ARM Cortex-M, orientada a dispositivos embebidos y con prestaciones óptimas para el IoT y las WSN.

### 3.4. Emulación del Diseño

Para validar la selección de protocolos y sistema operativo, será necesario realizar una emulación antes de comenzar con el diseño electrónico y la fabricación. Cada proyecto en Contiki OS dispone de un fichero de configuración, donde a través instrucciones al compilador (variables tipo define y macros), se seleccionan los distintos protocolos disponibles a emplear en cada nivel de la pila TCP/IP. De esta forma, solo se cargará el código correspondiente a la implementación seleccionada, reduciendo notablemente la carga en memoria flash necesaria.

#### 3.4.1. Configuración de Contiki OS

En Contiki OS, cada nivel de la pila TCP/IP está definido por la variable NETSTACK\_NAME, donde NAME hace referencia a cada nivel. A través de distintos ficheros con extensión .h, se consigue una asignación como la que se muestra en la Figura 18. En este formato de definición de protocolos, aparecen dos nuevos conceptos.

Por un lado, aparece el nivel RDC (Radio Duty Cycling). Este nivel es un nexo entre el nivel de enlace y el nivel físico, pensado para definir un intervalo de tiempo durante el cual la circuitería correspondiente a la transmisión y recepción radio permanecerá apagada. Esto permite obtener un mayor ahorro energético y alargar la batería de los dispositivos. En el caso que concierne a este proyecto no se utilizará nivel RDC. Por otro lado, en Contiki OS se realiza una separación a nivel de enlace entre la capa MAC y la capa que da formato al mensaje a nivel de enlace (Framer).

```

/* Network setup for IPv6 */
#define NETSTACK_CONF_NETWORK      sicslowpan_driver
#define NETSTACK_CONF_MAC         csma_driver
#define NETSTACK_CONF_RDC         nullrdc_driver
#define NETSTACK_CONF_RADIO       cooja_radio_driver
#define NETSTACK_CONF_FRAMER      framer_802154

#define NETSTACK_NETWORK           NETSTACK_CONF_NETWORK
#define NETSTACK_                  MAC NETSTACK_CONF_MAC
#define NETSTACK_RDC               NETSTACK_CONF_RDC
#define NETSTACK_RADIO             NETSTACK_CONF_RADIO
#define NETSTACK_FRAMER            NETSTACK_CONF_FRAMER

```

Figura 18 Asignación de protocolos en capa TCP/IP del dispositivo. Contiki OS

Cada nivel queda definido por un driver. Cada driver se define mediante la dupla compuesta por un fichero con extensión .c y un fichero con extensión .h. En este último se recogen las funciones que van a definir cada protocolo y en el fichero con extensión .c se recoge el código correspondiente a cada función.

En la Figura 19 se recoge el ejemplo de la configuración y drivers a emplear para los niveles físico, enlace, y red donde se incluyen las capas de adaptación 6LoWPAN, de red IPv6 y la capa de enrutamiento RPL. El nivel de transporte y de aplicación se definen sobre el fichero con extensión .c que define el proceso de aplicación.

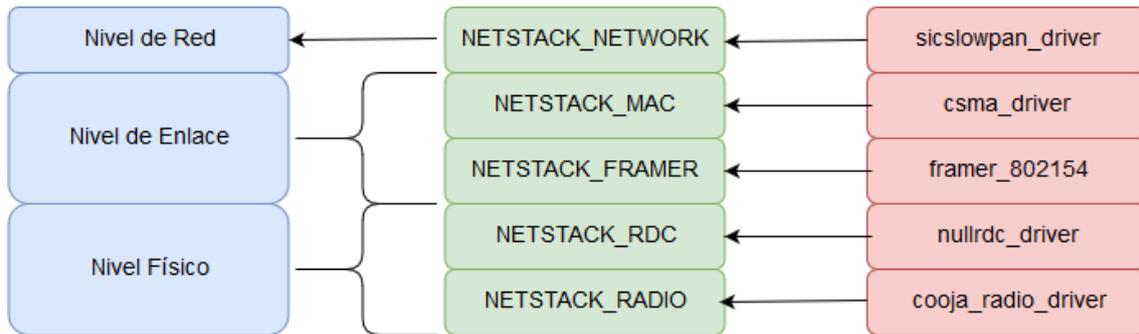


Figura 19 Asignación a nivel de driver para pila TCP/IP. Contiki OS

### 3.4.2. Entorno de Emulación Cooja

Una vez configurado el código para que la pila de protocolos coincida con nuestro diseño, a excepción del nivel de aplicación que será desarrollado de forma propia, queda validar la implementación. Para ello, Contiki OS ofrece una plataforma de emulación llamada Cooja [18]. Esta herramienta permite compilar un conjunto de ficheros con extensión .c sobre un nodo virtual de forma que cada nodo ejecute el conjunto de procesos definidos y poder así analizar el comportamiento en un entorno emulado.

Además, ofrece una herramienta de análisis de mensajes capturados a través del interfaz radio emulado. Este fichero obtenido dispone de extensión .pcap y podrá ser posteriormente analizado con herramientas de análisis de tráfico como Wireshark.

Se ha llevado a cabo una emulación de 10 nodos clientes con la configuración descrita en el apartado anterior que actualizan de forma constante la información a un nodo central o concentrador (servidor). El entorno de emulación que ofrece Cooja se muestra en la Figura 20. El nivel de aplicación es muy sencillo, donde los clientes envían un mensaje sonda de forma periódica con el contenido “Hello from the client”. Para ello se abre un socket UDP sobre los puertos 8765 en los clientes para transmisión y 5678 en el nodo concentrador para recepción.

En la Figura 21 se recoge el intercambio de mensaje correspondiente a un envío entre el nodo 10 y el nodo 1 (nodo concentrador). Dado que el nodo 10 no dispone de alcance directo hasta el nodo 1, busca a través del protocolo RPL, establecer una ruta hasta el nodo 1. En este caso de estudio, el nodo 10 utiliza el nodo 4, quien reenvía el mensaje hasta el nodo 1 al disponer de alcance radio directo. Por lo tanto, a nivel de enlace el camino seguido es 10-4-1. De esta forma confirmamos que nuestro sistema utiliza correctamente el protocolo RPL permitiendo el reenvío de mensajes y el encaminamiento.

Además, en la Figura 21 se puede observar como el tráfico capturado muestra el uso de los estándares IEEE 802.15.4 a nivel de enlace, 6LoWPAN como capa de adaptación para IPv6 como nivel de red. Del mismo modo, se ha conseguido configurar las rutas gracias al intercambio de mensajes RPL y se utiliza protocolo UDP como nivel de transporte. El análisis de cada uno de estos niveles se encuentra desarrollado en el Anexo D de esta memoria. En este caso de emulación, el nivel de aplicación no sigue ningún protocolo o estándar sino que es una cadena de caracteres. El protocolo CoAP será desarrollado e implementado posteriormente.

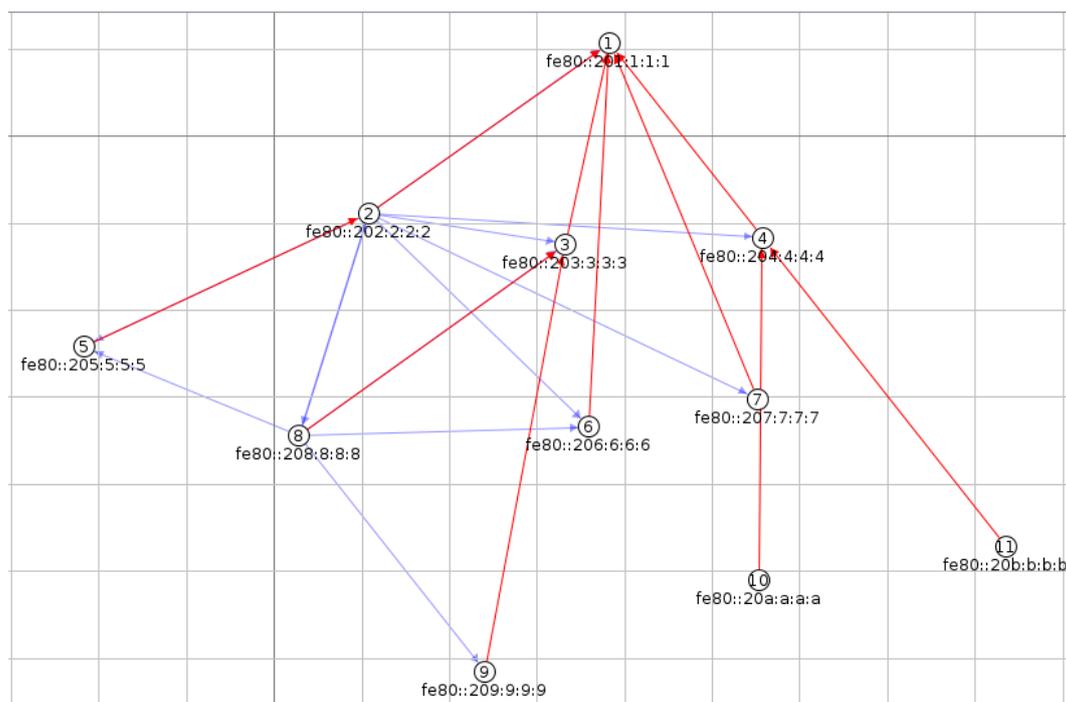


Figura 20 Ejemplo de red simulada. 10 nodos simples y 1 nodo concentrador

```
182 01:17.321 . . 66: 15.4 D 00:0A:00:0A:00:0A:00:0A 00:04:00:04:00:04:00:04|IPHC|IPv6|UDP
183 01:17.324 . . 3: 15.4 A
184 01:17.326 . . 75: 15.4 D 00:04:00:04:00:04:00:04 00:01:00:01:00:01:00:01|IPHC|IPv6|UDP
185 01:17.329 . . 3: 15.4 A
```

**IEEE 802.15.4 DATA #174**

From 0xABCD/00:0A:00:0A:00:0A:00:0A to 0xABCD/00:04:00:04:00:04:00:04

Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr = Long

**IPHC HC-06**

TF = 3, NH = inline, HLIM = 64, CID = 1, SAC = stateful, SAM = 3, MCast = false, DAC = stateful, DAM = 2

Contexts: sci=0 dci=0

**IPv6 TC = 0, FL = 0**

From aaaa:0000:0000:0000:000a:000a:000a:000a to aaaa:0000:0000:0000:0000:0000:0000:0001

**UDP**

Src Port: 8765, Dst Port: 5678

**Payload (25 bytes)**

```
001F56C0 48656C6C 6F203120 66726F6D 20746865 ..V.Hello 1 from the
20636C69 65 clie
```

**IEEE 802.15.4 DATA #7**

From 0xABCD/00:04:00:04:00:04:00:04 to 0xABCD/00:01:00:01:00:01:00:01

Sec = false, Pend = false, ACK = true, iPAN = true, DestAddr = Long, Vers. = 1, SrcAddr = Long

**IPHC HC-06**

TF = 3, NH = inline, HLIM = inline, CID = 1, SAC = stateful, SAM = 1, MCast = false, DAC = stateful, DAM = 2

Contexts: sci=0 dci=0

**IPv6 TC = 0, FL = 0**

From aaaa:0000:0000:0000:020a:000a:000a:000a to aaaa:0000:0000:0000:0000:0000:0000:0001

**UDP**

Src Port: 8765, Dst Port: 5678

**Payload (25 bytes)**

```
001F56C0 48656C6C 6F203120 66726F6D 20746865 ..V.Hello 1 from the
20636C69 65 clie
```

Figura 21 Ejemplo de intercambio de mensajes entre nodos 10 y 1 a través de nodo 4

### 3.5. Diseño Electrónico

Después de validar el correcto funcionamiento de la pila de protocolos seleccionada mediante la herramienta de emulación Cooja que ofrece Contiki OS, el último paso será realizar un diseño electrónico que permita la implementación mencionada.

En esta primera versión de diseño electrónico se trabajará con alimentación externa. Para ello será necesario realizar el diseño de una fuente de alimentación que permita adaptar la tensión de la red eléctrica al rango de tensiones empleado por la microelectrónica.

Posteriormente, aquellos nodos que conformen un dispositivo concentrador, necesitarán acceso a Internet mediante comunicación 2G o 3G tal y como se ha explicado en la memoria previamente. Por ello, será necesario disponer de un chipset que implemente estas tecnologías y permita realizar la comunicación a través de la red de telefonía móvil.

Finalmente, será necesario diseñar una tercera PCB que proporcione el interfaz radio en banda ISM a 868 MHz. Además, necesitará existir conexión tanto del chipset GSM como del transceptor radio con un microcontrolador, donde se encontrará en ejecución el sistema operativo Contiki OS encargado de gestionar los distintos procesos que intervienen en la red de comunicaciones.

#### 3.5.1. Fuente de Alimentación

Como se ha descrito anteriormente, existe la necesidad de realizar el diseño de una fuente de alimentación que permita adaptar los rangos de tensión para que pueda funcionar correctamente la microelectrónica como el transceptor radio, el chipset GSM o el propio microcontrolador.

Para ello se ha llevado a cabo el diseño de una fuente de alimentación flyback, cuyo objetivo es convertir los 230 V<sub>RMS</sub> de la red eléctrica en una salida constante de 5 V que alimentará a las dos PCBs conectadas a continuación. Esta PCB será denominada a lo largo de esta memoria como PCB1.

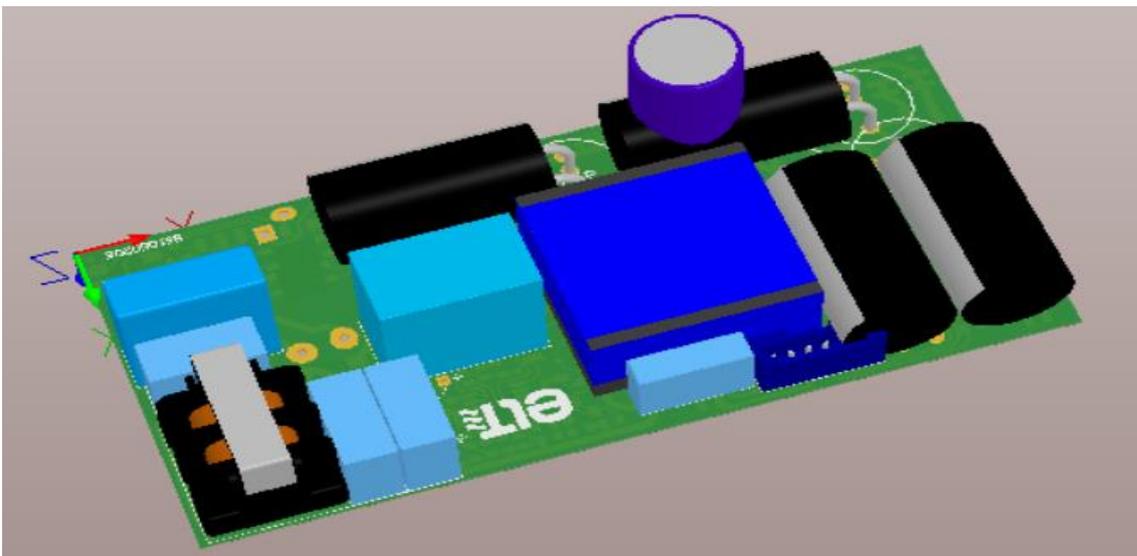


Figura 22 Representación 3D de PCB1. Fuente de alimentación

3.5.2. Placa Conexión 2G/3G

En segundo lugar, se ha llevado a cabo el diseño de la PCB que proporcione acceso a la red de telefonía móvil. Este diseño está pensado para los nodos concentradores que dispondrán de acceso a Internet. El diseño cuenta con un chipset que proporciona conectividad 2G y 3G. Para el correcto funcionamiento del dispositivo, será necesario utilizar una tarjeta SIM. Por ello, se ha realizado el montaje de un SIM holder en la parte posterior de la placa.

El chipset utiliza un interfaz UART para comunicación con el microcontrolador. El intercambio de información, el envío de órdenes o de programación se realiza a través de comandos AT, característicos de los módulos GSM. En la Figura 23 se muestra una representación 3D mediante las vistas alzado y posterior del diseño llevado a cabo.

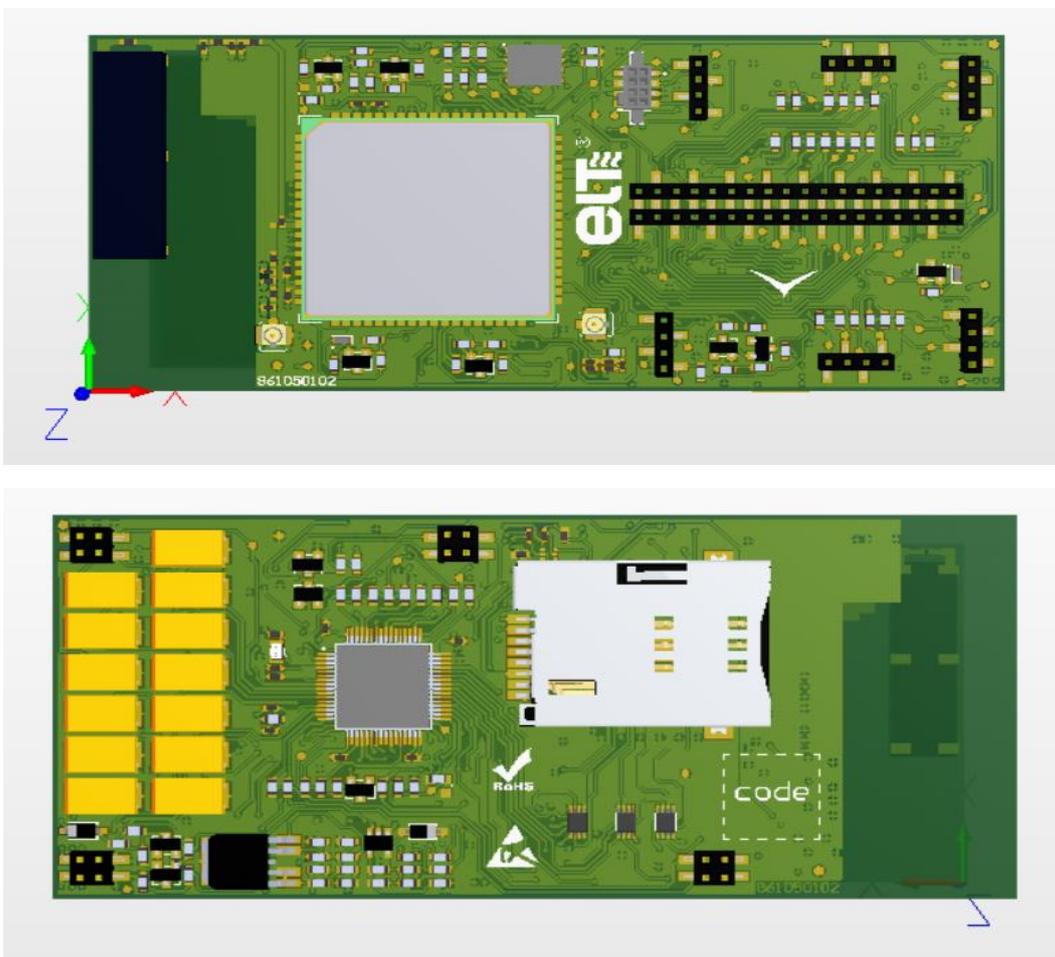


Figura 23 Representación 3D de la PCB2. Conexión 2G-3G

Para la selección de antena, en este caso se ha optado por una antena formato chip, colocada en un extremo de la PCB. En la parte posterior de la PCB se ha dejado la huella del microcontrolador y las pistas conectadas con dicha huella correctamente para poder realizar una implementación de dispositivo concentrador sin transceptor radio. Este tipo de dispositivos está orientado a aplicaciones donde con un único dispositivo sea suficiente para realizar la sensorización. Esta PCB será denominada a lo largo de esta memoria como PCB2.

### 3.5.3. Placa Microcontrolador y Transceptor Radio

Por último, la tercera PCB que compone el sistema, dispone del microcontrolador sobre el cuál se compilará el sistema operativo Contiki OS y donde se encontrará transceptor radio. El microcontrolador seleccionado es un ARM Cortex-M.

El transceptor radio dispone de un encapsulado VQNF con una potencia máxima de salida en bordes de antena de 14 dBm, apropiado para nuestro caso, donde la potencia de transmisión queda limitada a este mismo valor bajo la norma EN 300-220 que rige la normativa de espectro y compatibilidad electromagnética en la banda de 868 MHz (ver Anexo A). Puede configurarse una tasa de transmisión de datos desde 1.2 kbps hasta 200 kbps, pudiéndose implementar distintas modulaciones como ASK, PSK o GFSK. Esto permite cubrir cualquiera de las posibles implementaciones del nivel físico del estándar IEEE 802.15.4.

En referencia al consumo, se trata de un transceptor de bajo consumo con una corriente máxima consumida de 40 mA en transmisión y 0.5 mA en modo standby. El transceptor utiliza un interfaz SPI para comunicación con el microcontrolador. A través de dicho interfaz podemos editar cada uno de los registros de configuración que dispone, con el objetivo de configurar parámetros tales como la potencia de transmisión, la frecuencia central o portadora, la tasa de transmisión de datos, la modulación, el uso de codificación Viterbi y otros. Esta PCB será denominada a lo largo de esta memoria como PCB3.

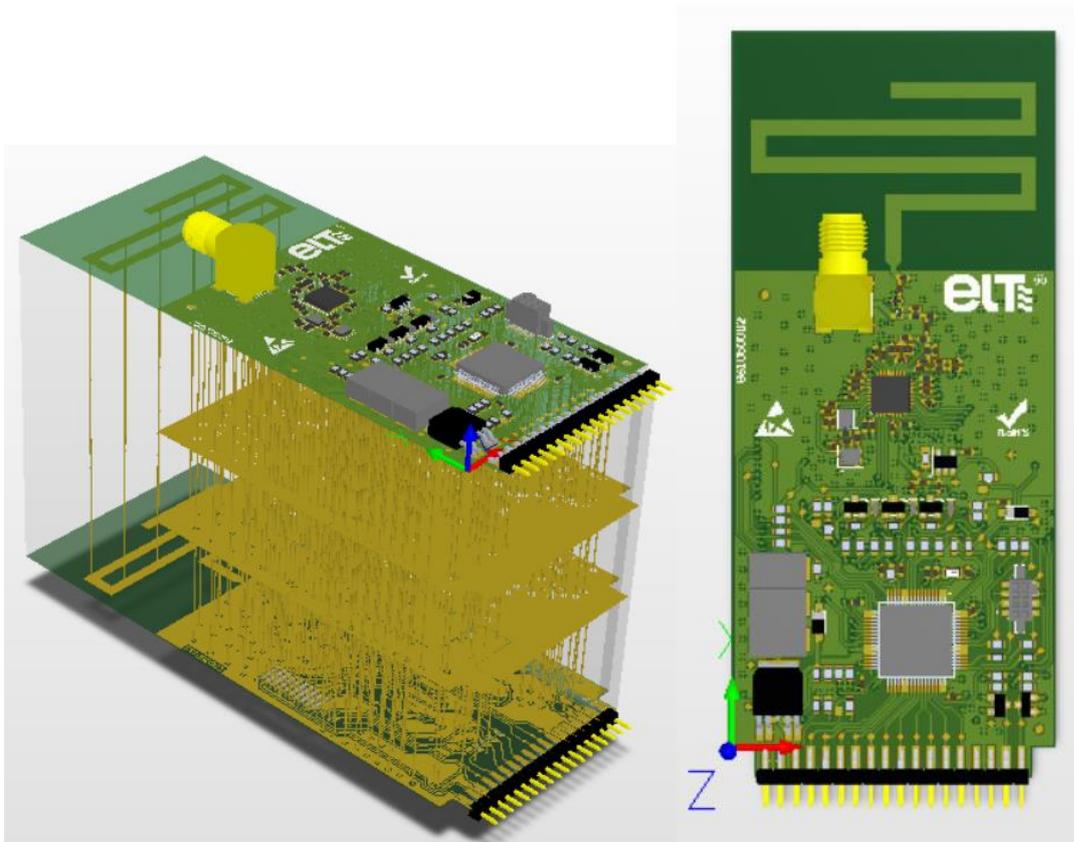


Figura 24 Representación 3D de la PCB3. Microcontrolador y transceptor radio

El ensamblaje de las tres PCB diseñadas, conformarán un diseño conjunto electrónico que irá protegido por una carcasa como la que se muestra en la Figura 25. El material de la carcasa ha sido seleccionado para no afectar a la radiopropagación del transceptor en la banda de 868 MHz. La base de dicha carcasa, una vez ensamblada la electrónica en su interior, es protegida por una capa de resina que garantiza un grado de protección IP67.



Figura 25 Dispositivo final cubierto por carcasa y enresinado

### 3.6. Procesos de Fabricación y Test

Una vez disponibles las PCB diseñadas, el siguiente paso es llevar a cabo el montaje de los componentes y obtener así un producto final para su posterior ensamblaje y validación. Para ello, se han utilizado las instalaciones de la empresa ELT (Especialidades Luminotécnicas S.A), que dispone de maquinaria para el montaje de componentes y diseño electrónico. En esta preserie se ha llevado a cabo la fabricación de 48 PCB1, 36 PCB2 (tan sólo 6 de ellas con el chipset 2G/3G soldado) y 36 PCB1.

#### 3.6.1. Proceso de Fabricación

El proceso de fabricación electrónica se ha llevado a cabo a través de dos tipos de montaje claramente diferenciados. En primer lugar, un proceso de soldadura de refusión, donde se realiza la inserción y el montaje de los componentes más pequeños y los circuitos integrados. Por otro lado, un proceso de soldadura manual para los componentes axiales y de mayor tamaño como transformadores o condensadores electrolíticos que se encuentran en la PCB1.

En referencia al proceso de refusión, tras aplicar pasta de soldadura sobre los pads donde los dispositivos han de ser soldados en la PCB, se colocan los componentes y se introduce en una máquina de refusión denominada reflow oven. Dicha máquina aplica distintas fases de calentamiento sobre la PCB, lo que permite que se produzca la fusión de la pasta de soldadura y de esta forma, el anclaje de los componentes electrónicos.

El proceso de soldadura manual es realizado por los operarios de la empresa ELT, cuyo resultado dispone de dos fases de verificación. En primer lugar, una revisión visual de las soldaduras y posteriormente a través de los procesos de test diseñados que se recogen en el siguiente apartado.

Tras finalizar el proceso completo de fabricación, las PCB2 que disponen del chipset 2G/3G (ver Figura 26), no son capaces en primera instancia de conectarse a la red de telefonía móvil. Ante esto, se barajan varias hipótesis como una mala comunicación a través del interfaz de comunicaciones con el microcontrolador, un fallo en el montaje de la antena o un fallo de soldadura en el propio chipset.

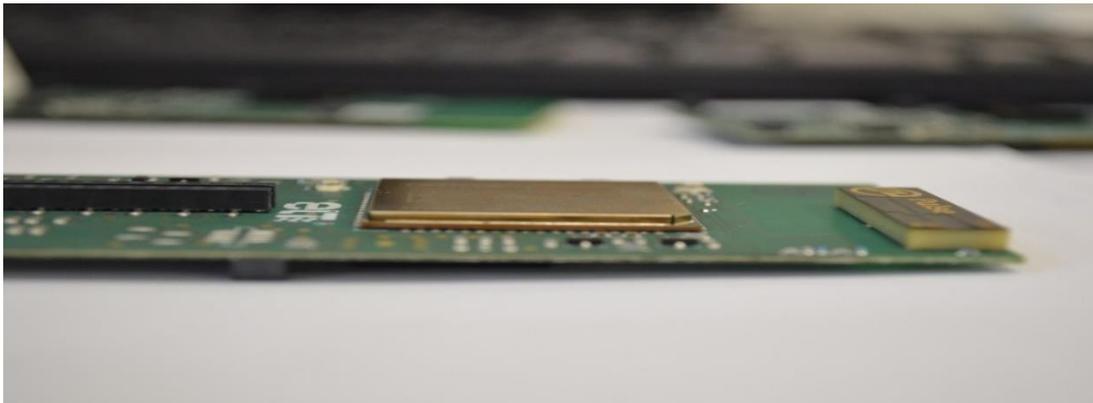


Figura 26 PCB2. Chipset 2G/3G y antena formato chip

Después de corroborar que las dos primeras hipótesis no son ciertas, se realiza un escáner de rayos X sobre el chipset 2G/3G. Como se puede apreciar en la Figura 27, los pads más externos del dispositivo presentan burbujas de mala fusión de la pasta de soldadura. Esto provocaba una alteración en la impedancia de algunos de los pines lo que conllevaba un mal funcionamiento de la electrónica. Finalmente se realiza la sustitución del modelo de pasta de soldadura utilizada y con ello se consigue con éxito la comunicación de los dispositivos a través de la red de telefonía móvil.

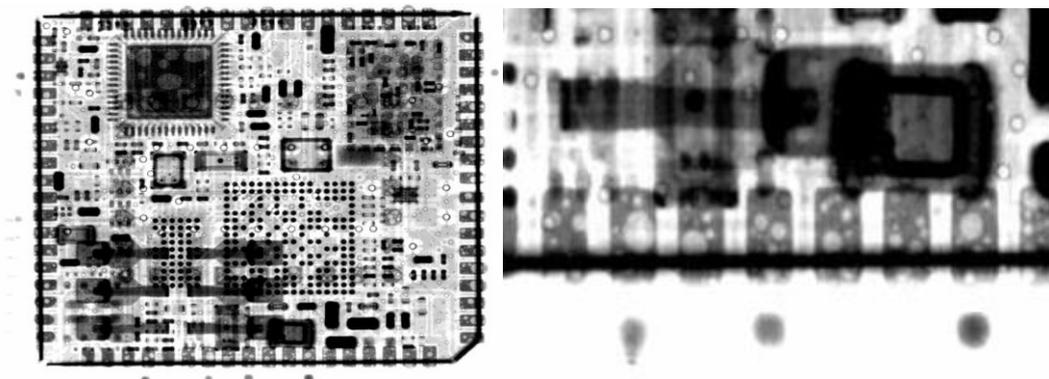


Figura 27 Escáner rayos X sobre chipset 2G/3G. Burbujas en la soldadura de los pads

### 3.6.2. Proceso de Test. Gestión de Fabricación Mediante ERP

Un ERP es un sistema de planificación de recursos empresariales. Permite gestionar la producción, logística, inventario, facturas, etc. de una compañía de forma modular. Con el objetivo de generar un proceso de fabricación eficiente, se ha diseñado una herramienta

para el grabado de microcontrolador y configuración del transceptor radio así como el desarrollo de una pequeña aplicación en JAVA que permite comunicar mediante puerto serie el ERP de la compañía ELT con la herramienta diseñada.

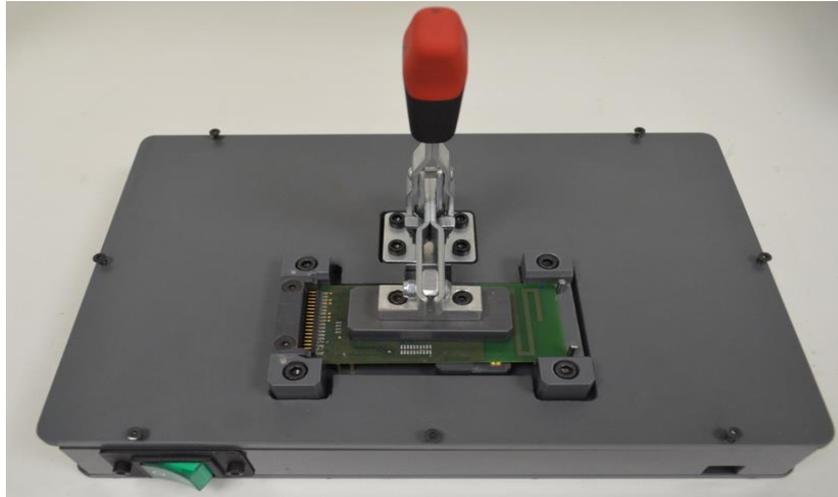


Figura 28 Útil diseñado para grabado y test de PCB3.

Esta herramienta fabricada (ver Figura 28) dispone de una PCB3 en su interior idéntica a la mostrada en el apartado 3.5.3, cuyo transceptor radio se encuentra esperando recibir mensajes de forma constante. También dispone de un conversor UART-USB que permite abrir una comunicación serie entre el microcontrolador y el PC, de forma que pueda enviar los resultados del test una vez finalizados.

El útil realizará el grabado de firmware a través de puerto de grabación sobre el microcontrolador disponible en la PCB3 a testear. Se configurará el transceptor radio y se procederá a testearlo mediante el intercambio de mensajes. Se chequeará el intercambio de mensajes de test y la potencia medida por la placa colocada en el interior.

Esto permitirá al operario encargado de la producción y fabricación obtener un resultado acerca del proceso de fabricación de la PCB3. Se han definido un total de 3 test donde se valida el grabado en memoria del microcontrolador, distintos periféricos, la alimentación y la configuración y envío a través del transceptor radio reportando el RSSI (Received Signal Strength Indicator) en dBm obtenido.

En la Figura 29 se muestran dos ejemplos de resultado de test para una PCB3.

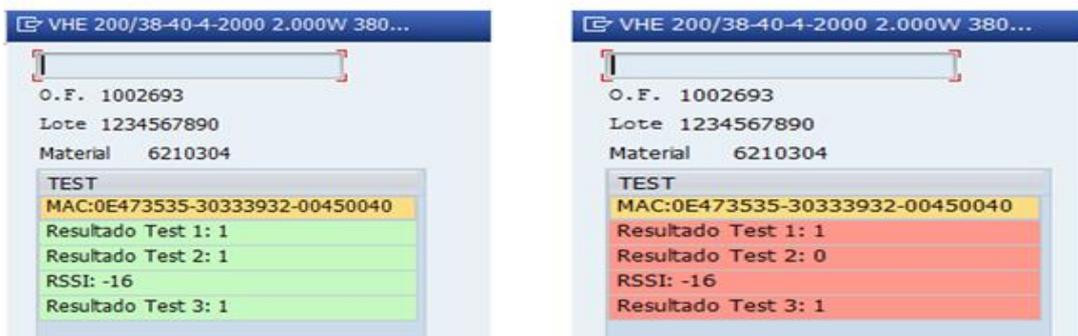


Figura 29 Ejemplos sobre ERP de grabado y test de PCB

3.6.3. Proceso de Test. Diseño de Herramienta Estadística

Para disponer de un proceso de calidad eficiente, se ha desarrollado un módulo en código Octave de representación gráfica y estadística. Esto permite a los encargados de los procesos de calidad obtener información referente a la fabricación de este diseño electrónico de forma visual y directa.

Se trata de un módulo almacenado en el ERP de ELT, ejecutado de forma automática tras finalizar una orden de fabricación. Recoge los principales parámetros durante el proceso de test para las PCB1, PCB2 y PCB3 diseñadas como el RSSI medido durante el test de las PCB3 de los transceptores, las tensiones de salida de las fuentes de alimentación (PCB1) o incluso el tiempo transcurrido entre el test de una placa y la siguiente.

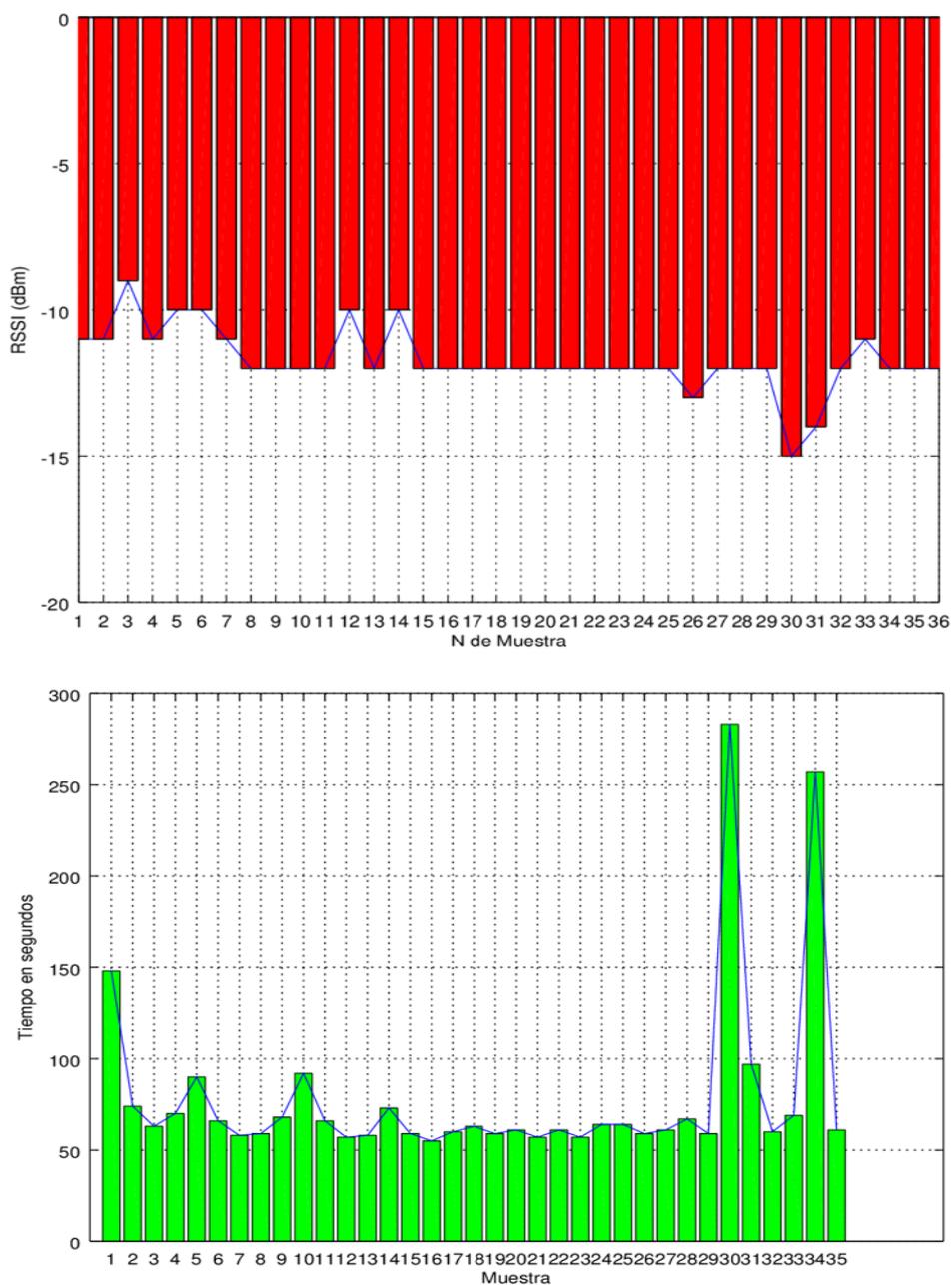


Figura 30 Resultados estadísticos de potencia y tiempo de test para PCB3

En la Figura 30 se muestra de forma gráfica los resultados tras la ejecución de este módulo diseñado al finalizar la fabricación de las PCB3. Se puede apreciar como los niveles de potencia medida en el útil son constantes (entre -9 y -15 dBm) siempre por encima del umbral mínimo para garantizar el correcto funcionamiento de la PCB3 durante el test (-20 dBm). El tiempo promedio de ejecución de test para cada PCB3 es de 79 segundos.

Del mismo modo para la PCB1, se muestra en la Figura 31 las tensiones de salida monitorizadas mediante su correspondiente útil. En este caso, de las 48 PCB1 fabricadas, 2 de ellas mostraban tensión de salida nula y fueron desechadas. Una tercera PCB1 (nº 11 en la Figura 31 identificada en color rojo) fue desechada al monitorizar una tensión de salida inferior al umbral (4.5 V). Los encargados del proceso de fabricación pudieron apreciar defectos en el transformador, evitándose de esta forma la salida al mercado de un producto defectuoso.

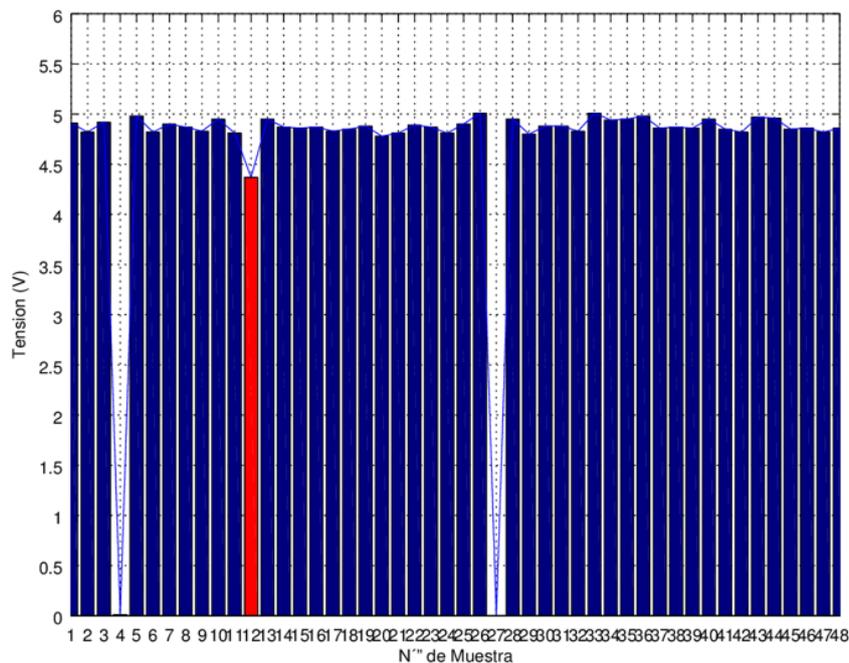


Figura 31 Resultados de tensión de salida para fuente de alimentación de 5V. PCB1

### 3.7. Pruebas y Validación

#### 3.7.1. Validación del Firmware sobre Microcontrolador

Disponibles el diseño electrónico una vez fabricado y superados los procesos test, el siguiente paso es realizar el grabado del firmware sobre el dispositivo. Para ello, el siguiente paso consiste en compilar Contiki OS sobre el microcontrolador situado en la PCB3. En primer lugar se lleva a cabo una asociación entre las fuentes de reloj del microcontrolador y las fuentes de tiempo del sistema operativo. Todas las fuentes de control y gestión de tiempo de Contiki OS son alimentados por la fuente de reloj definida con la estructura `current_clock`.

En este caso se ha utilizado el periférico System Tick Timer (SYSTICK). Este periférico se encuentra en ciertos dispositivos con arquitectura ARM. El SYSTICK utiliza la fuente

de reloj principal de la CPU (HCLK) para, en base a una configuración predeterminada, generar una interrupción de forma periódica. En este caso se ha configurado cada milisegundo de forma que la fuente de reloj de Contiki OS tendrá esta granularidad.

En la Figura 32 se encuentra el código desarrollado que vincula ambas fuentes de reloj, microcontrolador y sistema operativo.

```
void clock_it_handler(void) {
    current_clock++;
    if(etimer_pending() && etimer_next_expiration_time() <= current_clock) {
        etimer_request_poll();
        #if 0 //debug
        printf("%d,%d\n", clock_time(), etimer_next_expiration_time());
        #endif //debug
    }

    if (--second_countdown == 0) {
        current_seconds++;
        TOGGLE_PINDEBUG_PA0();
        second_countdown = CLOCK_SECOND;
    }
}

void SysTick_Handler(void)
{
    Timerlms++; //Timerlms_aux++;
    clock_it_handler();
}
```

Figura 32 Código vinculación de fuentes de reloj entre microcontrolador y S.O

También se ha realizado un estudio exhaustivo del código en referencia al conjunto de procesos de la pila TCP/IP que intervienen así como de la configuración del algoritmo RPL sobre Contiki OS. Se ha implementado una modificación sobre la métrica que ofrece el algoritmo RPL del sistema operativo (métrica muy simple basada en número de saltos).

En dicha implementación, la métrica se basa en el número de retransmisiones del mensaje esperadas (ETX) mediante un algoritmo de media móvil. De esta forma se consigue que el valor de la métrica obtenida en iteraciones anteriores tome mayor peso que las nuevas medidas de forma que la red sea más estable ante posibles fallos de comunicación puntuales. Esto ofrece mayor grado de estabilidad aunque el tiempo de adaptación y búsqueda de nueva ruta alternativa en caso de fallo definitivo en el enlace será mayor.

En el Anexo D de esta memoria se encuentran detallado el análisis de los procesos llevado a cabo así como una explicación del algoritmo modificado implementado.

### 3.7.2. Validación del Transceptor

Para validar el transceptor radio, se han realizado pruebas de radiación y alcance. En primer lugar, se ha obtenido los diagramas de radiación del diseño de la PCB3 que contiene el transceptor radio y la antena. Para dichas pruebas se ha utilizado un escáner de antenas.

Tras finalizar el ensayo se obtiene el diagrama de radiación mostrado en la Figura 33.

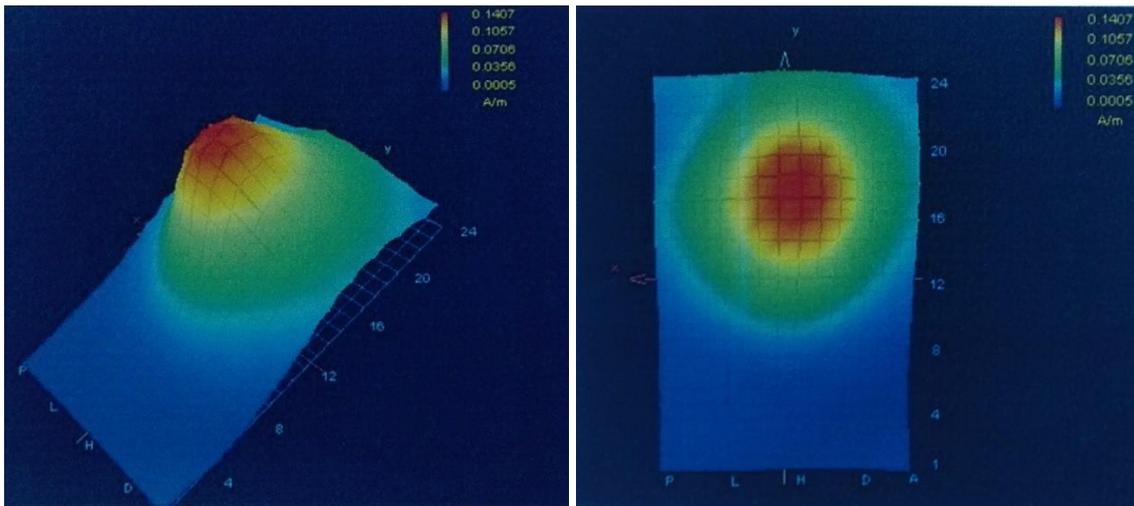


Figura 33 Vista 3D de la radiación. Plano X-Y

La antena presenta un diagrama de radiación prácticamente idéntico al de una antena monopolo omnidireccional. En la Figura 33 se muestra el diagrama de radiación de la antena donde se puede apreciar como la radiación es máxima en el eje XZ y prácticamente nula en la dirección Y.

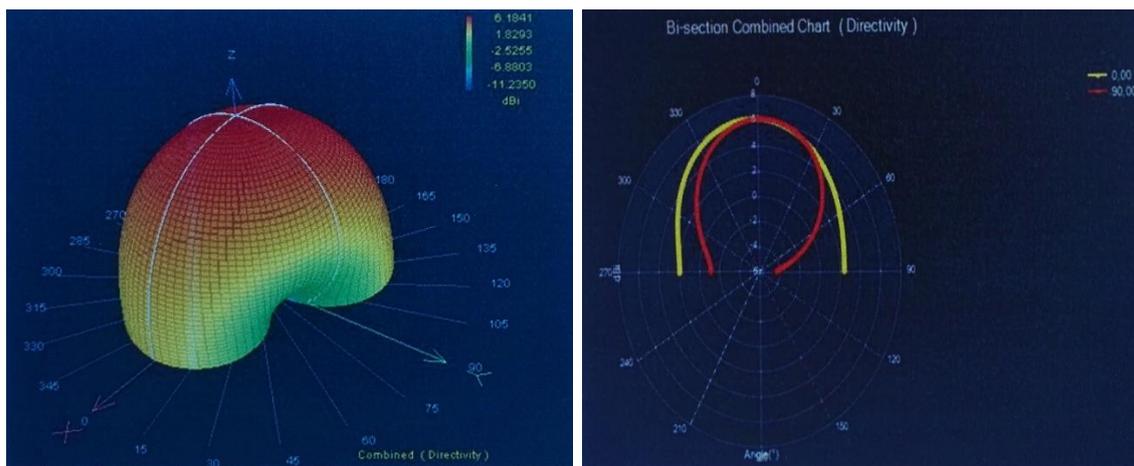


Figura 34 Vista 3D. Diagrama de radiación. Cortes del diagrama

Por otro lado, con el objetivo de validar el alcance del radioenlace a través de los transceptores, se ha realizado un conjunto de ensayos de alcance monitorizando el valor de potencia recibida a través del RSSI. Para dichos ensayos se ha utilizado el intercambio de mensajes entre dos placas PCB3, con desplazamiento de una de ellas cada 2 metros y visión directa. En la Figura 35 se muestran, en primer lugar, los resultados obtenidos tras la realización de 5 ensayos.

El objetivo de este ensayo es obtener una comparativa con la curva de propagación teórica (ver Anexo F). Como se puede apreciar en esta misma figura, la curva teórica se aproxima correctamente a la curva promedio obtenida de los ensayos realizados.

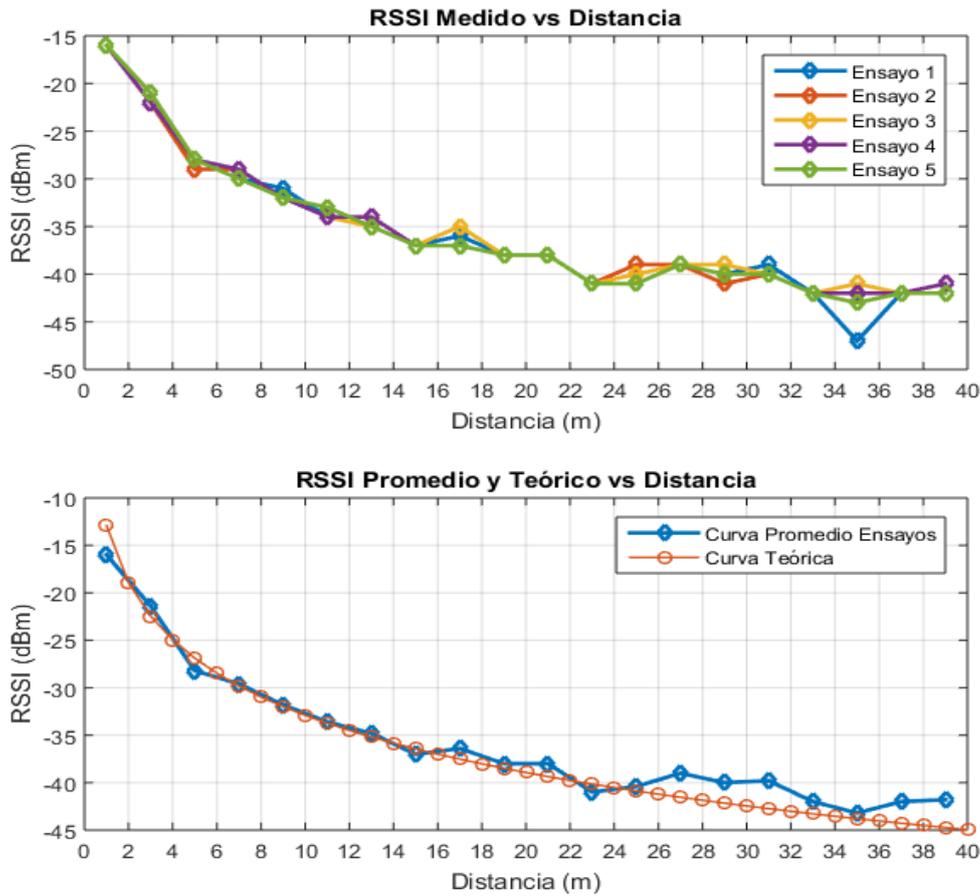


Figura 35 Curvas RSSI. Ensayos y promedio vs curva teórica

### 3.8. Implementación del Protocolo de Aplicación

Como se ha mencionado anteriormente, el protocolo de aplicación empleado es CoAP. Contiki OS ofrece una librería que permite definir el formato de los mensajes CoAP en lenguaje C a través de una variable tipo struct denominado `coap_packet_t`. Dicha estructura dispone de todos los campos que define el protocolo y será la variable que se pasará al nivel de transporte (UDP en este caso).

Contiki OS ofrece todo un conjunto de funciones para dar formato a un mensaje y simplificar el proceso de análisis tras la recepción de un mensaje en base al estándar. El resumen, el sistema operativo ofrece una API que simplifica el proceso de programación. En la Figura 36 se muestra un ejemplo con prototipo de funciones y definiciones que ofrece Contiki OS.

Sin embargo, esta librería no proporciona una definición de los recursos. Es decir, no define el formato, tamaño o acceso a las variables a consultar o actualizar. Por ello, se ha realizado la implementación de los recursos así como las funciones que ligán el tipo de mensaje CoAP (PUT, GET, POST o DELETE) con la actualización de los recursos.

```

int coap_get_header_block2(void *packet, uint32_t *num, uint8_t *more,
                           uint16_t *size, uint32_t *offset);
int coap_set_header_block2(void *packet, uint32_t num, uint8_t more,
                           uint16_t size);

int coap_get_header_block1(void *packet, uint32_t *num, uint8_t *more,
                           uint16_t *size, uint32_t *offset);
int coap_set_header_block1(void *packet, uint32_t num, uint8_t more,
                           uint16_t size);

int coap_get_header_size2(void *packet, uint32_t *size);
int coap_set_header_size2(void *packet, uint32_t size);

int coap_get_header_size1(void *packet, uint32_t *size);
int coap_set_header_size1(void *packet, uint32_t size);

int coap_get_payload(void *packet, const uint8_t **payload);
int coap_set_payload(void *packet, const void *payload, size_t length);

/* CoAP message types */
typedef enum {
    COAP_TYPE_CON,           /* confirmables */
    COAP_TYPE_NON,         /* non-confirmables */
    COAP_TYPE_ACK,         /* acknowledgements */
    COAP_TYPE_RST          /* reset */
} coap_message_type_t;

/* CoAP request method codes */
typedef enum {
    COAP_GET = 1,
    COAP_POST,
    COAP_PUT,
    COAP_DELETE
} coap_method_t;

```

Figura 36 Ejemplo de funciones prototipo y definiciones CoAP en Contiki OS

### 3.8.1. Implementación de Recursos CoAP.

Se ha definido la variable de tipo struct `resource_struct_t` que almacena la información referente a los recursos CoAP. Se han definido un total de 8 recursos, pensando en la posible conexión a múltiples dispositivos tipo sensor, puerto de comunicaciones etc.

Como se puede apreciar en la Figura 37, los recursos son de tipo `uint8_t`, es decir, son vectores de bytes. CoAP trabaja con codificación ASCII (American Standard Code for Information Interchange). En esta misma imagen se muestra como las URL a las que accede cada dispositivo son de tipo vector de caracteres. Es decir, cada carácter tiene su codificación ASCII que ocupa 1 byte.

```

typedef struct resources_struct{           static const char *service_urls[NUMBER_OF_URLS] =

    uint8_t id [UNIQUE_ID_SIZE];         "/0",
    uint8_t Resource0[SIZE_RES_0];       "/1",
    uint8_t Resource1[SIZE_RES_1];       "/2",
    uint8_t Resource2[SIZE_RES_2];       "/3",
    uint8_t Resource3[SIZE_RES_3];       "/4",
    uint8_t Resource4[SIZE_RES_4];       "/5",
    uint8_t Resource5[SIZE_RES_5];       "/6",
    uint8_t Resource6[SIZE_RES_6];       "/7",
    uint8_t Resource7[SIZE_RES_7];
    uint8_t next_resource;                };
}resources_struct_t;

```

Figura 37 Definición recursos CoAP en lenguaje C

Sin embargo, este sistema ha sido diseñado para trabajar con otros dispositivos o sensores cuya comunicación, como se ha descrito en apartados anteriores, suele realizarse a través de puertos digitales, donde el intercambio se realiza a nivel de bit. A modo de ejemplo, en el siguiente capítulo se recoge un despliegue de red donde el dispositivo es conectado a través de un bus de comunicaciones digital con un driver inteligente para alimentación de cargas LED en entornos de alumbrado público. Las especificaciones de éste se recogen en el Anexo H de esta memoria.

Dicho driver enviará al dispositivo toda la información capaz de recoger (decenas de bytes) con información referente a la tensión de red eléctrica, tensión de la carga LED, corriente suministrada etc. Toda esta información conforma un recurso CoAP, por lo que será mucho más costoso en términos de memoria y en tamaño de mensaje utilizar ASCII en vez de codificación a nivel de bit directamente. Además se evita de esta forma tener que realizar fragmentación de los mensajes a nivel 6LoWPAN, simplificando el envío y recepción de tramas.

Por ello se ha realizado una modificación sobre el estándar CoAP, y tanto el servidor como los dispositivos clientes interpretan el valor de consulta o de actualización como valores a nivel de bit.

### 3.9. Despliegue de Validación

El último paso antes de poder comercializar el producto desarrollado será validar en una instalación demo una implementación de la red de comunicaciones diseñada. Para ello, se realiza el despliegue de una red de dispositivos compuesta por 5 nodos simples y un nodo concentrador con acceso a Internet a través de tecnología 3G. El despliegue tiene lugar en las instalaciones de la empresa ELT. En la Figura 38 se muestra el plano con la distribución de los puntos de luz. Cada dispositivo es instalado sobre una luminaria de forma que la monitorización en este caso se realizará sobre un driver de alimentación de ELT para luminarias basadas en tecnología LED.

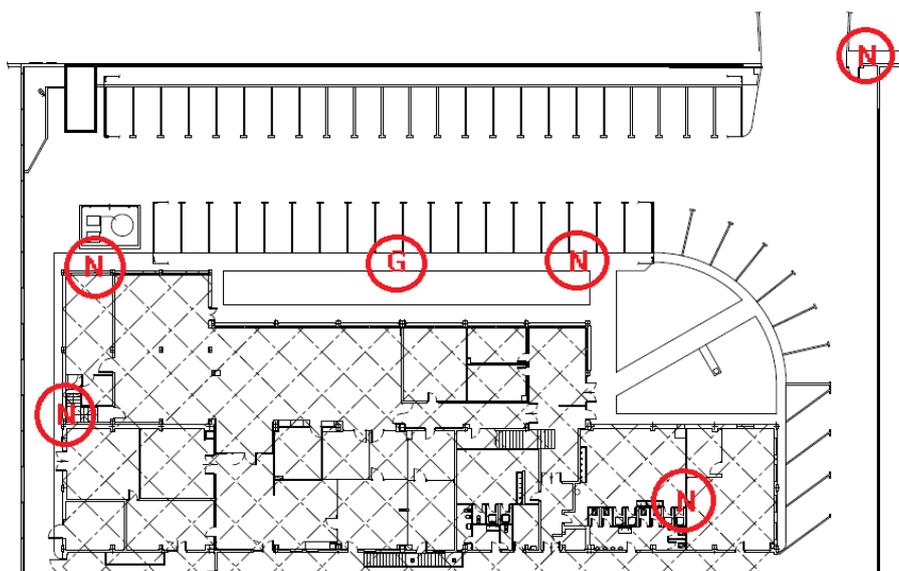


Figura 38 Mapa del despliegue en instalaciones de ELT

Los dispositivos actualizarán y consultarán información de sus recursos al servidor a través del nodo concentrador. La conexión se realiza con un servidor interno situado en las instalaciones de ELT donde corre un servidor CoAP que expone los recursos descritos en el apartado anterior.

Los dispositivos simples son colocados en el tejado de las instalaciones, en la entrada y en un departamento dentro de las propias instalaciones para validar la radiopropagación en un entorno más hostil. El dispositivo concentrador se coloca sobre el tejado. Las direcciones IPv6 globales unicast se obtienen utilizando el prefijo de red guardado en contexto (conocido por los nodos simples y concentradores). Dichas direcciones son utilizadas para el intercambio de mensajes CoAP entre los dispositivos y el servidor.

Para los mensajes RPL sobre ICMPv6, se utiliza el prefijo local FE80::. Esto es así dado que la propagación de estos mensajes sólo tiene sentido dentro de la propia red y no requieren interacción con el exterior. Como interface ID (últimos 64 bits de la dirección IPv6) siempre se utiliza la dirección MAC (formato Little-Endian) invirtiendo el universal/local bit [19].

La Tabla 3 muestra las direcciones físicas de cada dispositivo y la topología de éstos. En la Tabla 4 se muestra el mapeo de direcciones que realiza 6LoWPAN.

ID del Nodo	Dispositivo	Dirección MAC
00000001	Simple	00:45:00:2B:30:33:39:32
00000002	Simple	00:45:00:3A:30:33:39:32
00000003	Simple	00:3A:00:3F:30:33:39:32
00000004	Simple	00:4A:00:2C:38:33:36:35
00000005	Simple	00:3D:00:2B:38:33:36:35
00000006	Concentrador	00:3B:00:30:30:33:39:32

Tabla 3 Direcciones MAC y topología de los dispositivos instalados

Dirección MAC	IPv6 Link Local	IPv6 Global Unicast
00:45:00:2B:30:33:39:32	fe80::2900:4500:3239:3330	2560:af32:32de:131f:2900:4500:3239:3330
00:45:00:3A:30:33:39:32	fe80::3800:4500:3239:3330	2560:af32:32de:131f:3800:4500:3239:3330
00:3A:00:3F:30:33:39:32	fe80::3d00:3a00:3239:3330	2560:af32:32de:131f:3d00:3a00:3239:3330
00:4A:00:2C:38:33:36:35	fe80::2e00:4a00:3536:3338	2560:af32:32de:131f:2e00:4a00:3536:3338
00:3D:00:2B:38:33:36:35	fe80::1e00:4100:3536:3338	2560:af32:32de:131f:1e00:4100:3536:3338
00:3B:00:30:30:33:39:32	fe80::3200:3b00:3239:3330	2560:af32:32de:131f:ff:fe00:1

Tabla 4 Direcciones IPv6 de los dispositivos instalados

El proceso de instalación de los 6 puntos de luz con los dispositivos implantados llevo un total de 8 horas, realizado por la división de sistemas inteligentes de ELT. En la Figura 39 se muestran imágenes de la instalación llevada a cabo.



Figura 39 Imágenes de luminarias instaladas con dispositivo diseñado incorporado

Para poder verificar el comportamiento de la red, se ha diseñado una herramienta de ingeniería que permite monitorizar el tráfico de la red de comunicaciones dentro del alcance radio del transceptor. La aplicación ha sido desarrollada en lenguaje Java y permite realizar la conversión de los mensajes recibidos por un transceptor y enviados a un PC a través de un conversor UART/USB a un fichero con extensión hex\_dump que posteriormente pueda ser analizado con la herramienta de análisis de tráfico Wireshark. La aplicación se muestra en el Anexo E de esta memoria.

Como se puede apreciar en la Figura 40, se recopila el tráfico de los 6 dispositivos que intervienen en este despliegue. En esta misma figura se muestra una captura del tráfico aplicándose un filtro de protocolo CoAP para mostrar únicamente el tráfico referente al nivel de aplicación.

Como se puede apreciar, los 5 dispositivos simples realizan de forma sucesiva consulta a recursos mediante el comando GET y su correspondiente respuesta así como la actualización del recursos con el comando PUT. Se puede apreciar, tal y como se indicaba anteriormente, que los mensajes RPL utilizan las direcciones de enlace local utilizando el prefijo fe80:: y los mensajes de aplicación CoAP utilizan direcciones IPv6 globales unicast.

## CAPÍTULO 3. IMPLEMENTACIÓN DE LA RED DE COMUNICACIONES

En el Anexo I de esta memoria se recogen capturas del análisis de tráfico en forma de diagrama de comunicaciones donde se puede apreciar el intercambio de mensaje CoAP y RPL de los nodos que intervienen en el tráfico capturado.

The top screenshot shows a Wireshark capture with the following data:

No.	Time	Source	Destination	Protocol	Length	Info
271	634.000000			IEEE 802.15.4	3	Ack
270	634.000000	fe80::2e00:4a00:3536:3338	fe80::3800:4500:3239:3330	ICMPv6	100	RPL Control (DODAG Information Object)
269	628.000000			IEEE 802.15.4	3	Ack
268	628.000000	2560:af32:32de:131f:3d00:3a00:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:7, PUT
267	625.000000			IEEE 802.15.4	3	Ack
266	625.000000	fe80::3d00:3a00:3239:3330	fe80::3800:4500:3239:3330	ICMPv6	100	RPL Control (DODAG Information Object)
265	584.000000			IEEE 802.15.4	3	Ack
264	584.000000	2560:af32:32de:131f:0:ff:fe00:1	2560:af32:32de:131f:2900:4500:3239:3330	CoAP	77	ACK, MID:26, 2.04 Changed
263	584.000000			IEEE 802.15.4	3	Ack
262	584.000000	2560:af32:32de:131f:2900:4500:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	64	CON, MID:6, GET, //1
261	580.000000			IEEE 802.15.4	3	Ack
260	580.000000	2560:af32:32de:131f:2e00:4a00:3536:3338	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:7, PUT
259	559.000000			IEEE 802.15.4	3	Ack
258	559.000000	2560:af32:32de:131f:0:ff:fe00:1	2560:af32:32de:131f:3d00:3a00:3239:3330	CoAP	77	ACK, MID:26, 2.04 Changed
257	559.000000			IEEE 802.15.4	3	Ack
256	559.000000	2560:af32:32de:131f:3d00:3a00:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	64	CON, MID:6, GET, //1
255	559.000000			IEEE 802.15.4	3	Ack
254	550.000000	fe80::3800:4500:3239:3330	fe80::2900:4500:3239:3330	ICMPv6	100	RPL Control (DODAG Information Object)
253	534.000000			IEEE 802.15.4	3	Ack
252	534.000000	fe80::2e00:4a00:3536:3338	fe80::2900:4500:3239:3330	ICMPv6	100	RPL Control (DODAG Information Object)
251	532.000000			IEEE 802.15.4	3	Ack
250	532.000000	2560:af32:32de:131f:0:ff:fe00:1	2560:af32:32de:131f:1e00:4100:3536:3338	CoAP	77	ACK, MID:26, 2.04 Changed
249	532.000000			IEEE 802.15.4	3	Ack

The bottom screenshot shows a detailed view of CoAP messages with the following data:

No.	Time	Source	Destination	Protocol	Length	Info
43	63.000000	2560:af32:32de:131f:3d00:3a00:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:1, PUT
71	93.000000	2560:af32:32de:131f:3800:4500:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:1, PUT
74	101.000000	2560:af32:32de:131f:1e00:4100:3536:3338	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:1, PUT
78	102.000000	2560:af32:32de:131f:2900:4500:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:1, PUT
80	102.000000	2560:af32:32de:131f:2e00:4a00:3536:3338	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:1, PUT
101	153.000000	2560:af32:32de:131f:3d00:3a00:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	64	CON, MID:2, GET, //1
103	154.000000	2560:af32:32de:131f:0:ff:fe00:1	2560:af32:32de:131f:3d00:3a00:3239:3330	CoAP	77	ACK, MID:8, 2.04 Changed
111	176.000000	2560:af32:32de:131f:3800:4500:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	64	CON, MID:2, GET, //1
113	176.000000	2560:af32:32de:131f:0:ff:fe00:1	2560:af32:32de:131f:3800:4500:3239:3330	CoAP	77	ACK, MID:8, 2.04 Changed
115	183.000000	2560:af32:32de:131f:2900:4500:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	64	CON, MID:2, GET, //1
117	183.000000	2560:af32:32de:131f:0:ff:fe00:1	2560:af32:32de:131f:2900:4500:3239:3330	CoAP	77	ACK, MID:8, 2.04 Changed
119	184.000000	2560:af32:32de:131f:1e00:4100:3536:3338	2560:af32:32de:131f:0:ff:fe00:1	CoAP	64	CON, MID:2, GET, //1
121	184.000000	2560:af32:32de:131f:0:ff:fe00:1	2560:af32:32de:131f:1e00:4100:3536:3338	CoAP	77	ACK, MID:8, 2.04 Changed
127	217.000000	2560:af32:32de:131f:2e00:4a00:3536:3338	2560:af32:32de:131f:0:ff:fe00:1	CoAP	64	CON, MID:2, GET, //1
129	217.000000	2560:af32:32de:131f:0:ff:fe00:1	2560:af32:32de:131f:2e00:4a00:3536:3338	CoAP	77	ACK, MID:8, 2.04 Changed
134	243.000000	2560:af32:32de:131f:3d00:3a00:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:3, PUT
152	268.000000	2560:af32:32de:131f:1e00:4100:3536:3338	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:3, PUT
157	281.000000	2560:af32:32de:131f:2e00:4a00:3536:3338	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:3, PUT
161	283.000000	2560:af32:32de:131f:3800:4500:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:3, PUT
166	297.000000	2560:af32:32de:131f:2900:4500:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	109	CON, MID:3, PUT
174	341.000000	2560:af32:32de:131f:2e00:4a00:3536:3338	2560:af32:32de:131f:0:ff:fe00:1	CoAP	64	CON, MID:4, GET, //1
176	341.000000	2560:af32:32de:131f:0:ff:fe00:1	2560:af32:32de:131f:2e00:4a00:3536:3338	CoAP	77	ACK, MID:14, 2.04 Changed
178	346.000000	2560:af32:32de:131f:3d00:3a00:3239:3330	2560:af32:32de:131f:0:ff:fe00:1	CoAP	64	CON, MID:4, GET, //1

Figura 40 Capturas de tráfico en despliegue. Filtro por protocolo de aplicación CoAP

En la Figura 41 se muestra un análisis del tráfico en cantidad de mensajes por minuto. El intervalo de análisis es de 100 minutos. En la primera imagen se muestra el tráfico total mientras que en la segunda imagen se realiza una división del tráfico correspondiente al establecimiento y mantenimiento de la red a través del protocolo RPL (gráfica de color negro) y el tráfico correspondiente al protocolo de aplicación CoAP (gráfica color rojo).

Como es de esperar, existe mayor tráfico RPL al principio hasta la conformación de la red. Posteriormente, se observa cómo existe un pico en el tráfico RPL cada 1000 segundos correspondiente al envío de mensajes para mantenimiento de la red por parte del nodo concentrador. El tráfico CoAP se mantiene constante dado que el envío de mensajes de consulta y actualización de recursos se realiza con un tiempo aleatorio entre 1 y 2 minutos.

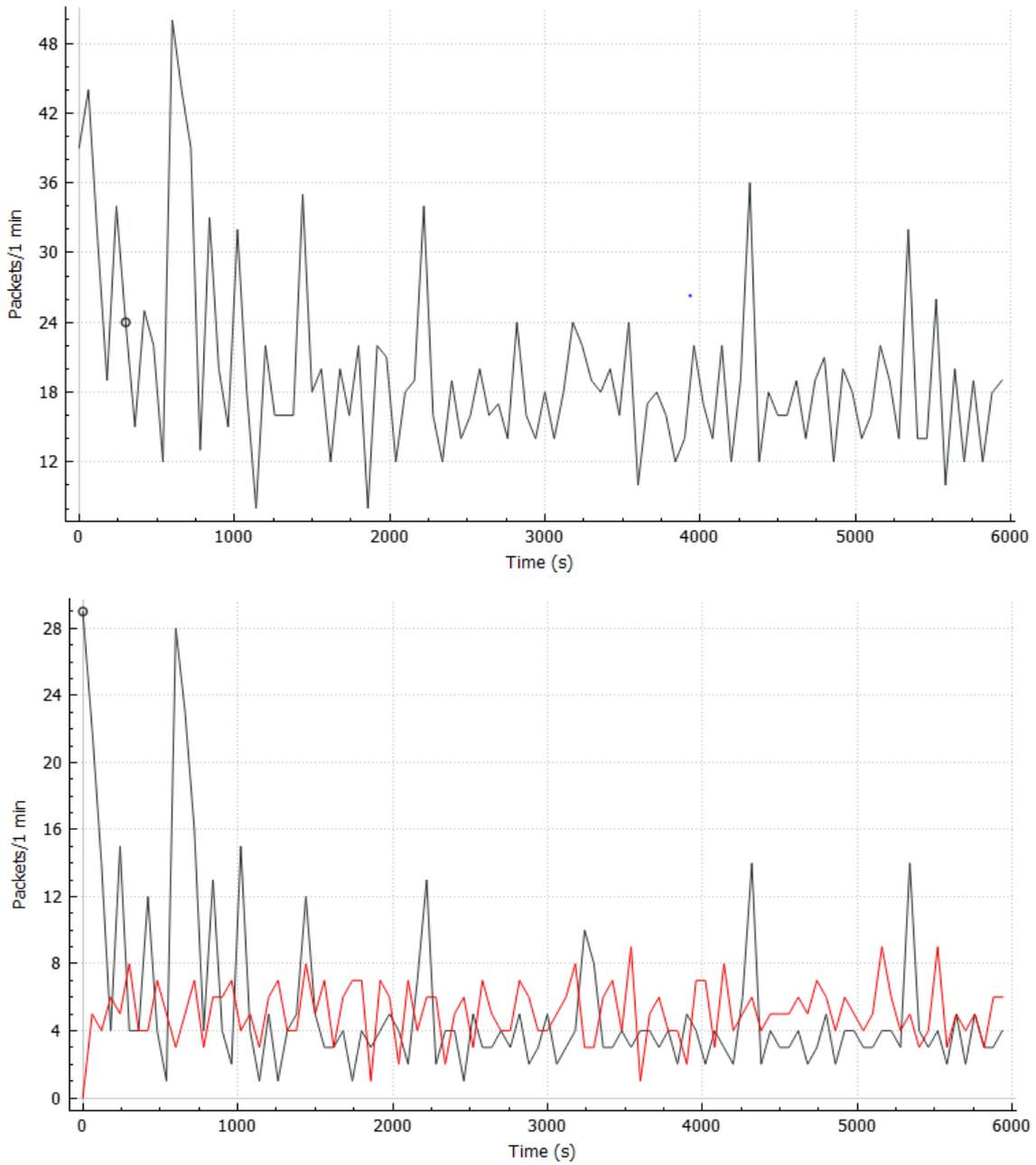


Figura 41 Análisis de tráfico. Tráfico RPL vs tráfico CoAP

## 4. Despliegue de la Red de Comunicaciones

Con el objetivo de validar la totalidad del diseño y la implementación, en este apartado se recogen los resultados de realizar una instalación y montaje en un proyecto de la empresa ELT. La división de sistemas inteligentes de ELT ha desarrollado STELARIA [20], una solución integral (hardware y software) en colaboración con otras empresas para la iluminación inteligente punto a punto en entornos urbanos que va mucho más allá del concepto tradicional de telegestión y que aplica la comunicación e interacción de los objetos a través de Internet.

STELARIA utiliza la red de comunicaciones desarrollada en esta memoria como sistema de acceso a Internet. En este caso se realizará una monitorización de los driver o fuente de alimentación (consumo, corriente aplicada sobre la carga LED, tensión, etc.) que acompañan a cada punto de luz. El objetivo es proporcionar al gestor y mantenedor del alumbrado información en tiempo real para el mantenimiento preventivo de las instalaciones, mediante la medida de parámetros relacionados con el propio driver, el módulo LED o la red eléctrica, posibilitando procedimientos de diagnóstico y test, y la gestión eficaz de alarmas e incidencias.

La monitorización y control del driver y de cada una de las luminarias se realizan mediante un paquete software propietario con distintos módulos y aplicaciones de gestión web, multi-dispositivo y con diferentes perfiles de usuario, basado en la representación geolocalizada de las luminarias. En el Anexo G de esta memoria se recoge más información acerca del sistema STELARIA.

### 4.1. Despliegue de la Red en Instalación

En este despliegue se lleva a cabo la instalación de un total de 41 luminarias incorporadas con los dispositivos desarrollados. En primer lugar, se realiza un estudio de cobertura basado en posicionamiento geográfico de cada punto de luz.

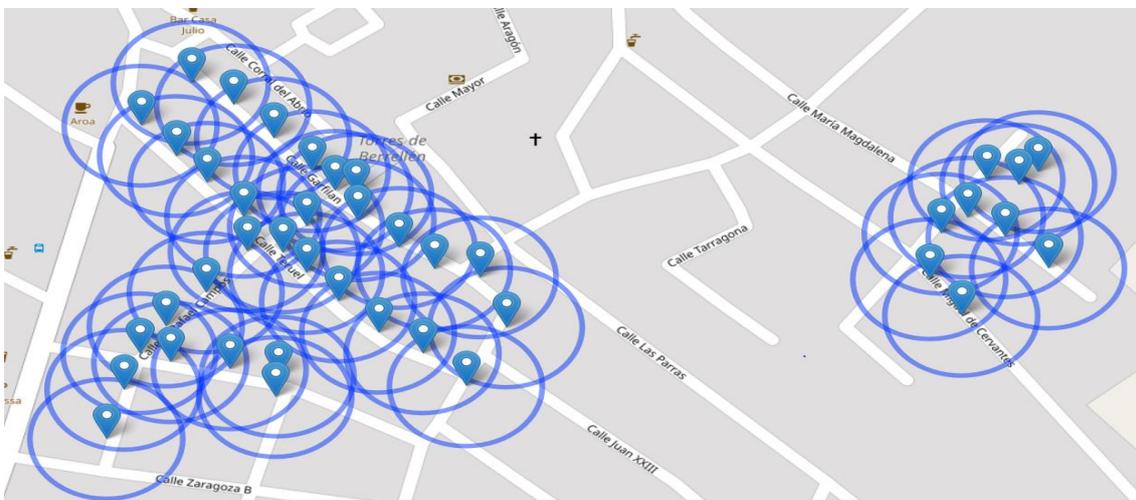


Figura 42 Mapa de despliegue en instalación con rangos de cobertura adjuntos

Si bien es cierto que la cobertura conseguida por el transceptor es superior al centenar de metros, para el cálculo de coberturas se ha utilizado una estimación de rango por dispositivo de 50 metros dejando un margen holgado. En la Figura 42 se muestra un plano con cada punto instalado y su rango de cobertura.

En primer lugar se garantiza que ninguno de los puntos quede aislado o no disponga de cobertura. En dicho caso sería necesario instalar un sistema repetidor. Por otro lado, una vez verificada esta primera premisa, se selecciona la ubicación de los dispositivos concentradores que dispondrán de acceso 3G en este caso. Para ello se seleccionan aquellos nodos que cubran la mayor parte posible de dispositivos con el menor número de saltos posible. Previamente se habrá validado la existencia de cobertura por el operador de telefonía móvil. En la Figura 43 se muestra una imagen referente al proceso de instalación llevado a cabo en este despliegue de proyecto STELARIA. La luminaria lleva incorporada el sistema desarrollado en este proyecto.



Figura 43 Imágenes sustitución de puntos de luz. Instalación de STELARIA

En este caso de despliegue, de los 41 puntos que conforman la red, 3 de ellos son concentradores con acceso a Internet y 38 son nodos simples. Cada dispositivo concentrador tiene capacidad de gestionar el tráfico de un total de 50 puntos, pero en este despliegue existen dos zonas aisladas (ver Figura 42), lo que implica utilizar al menos dos puntos de acceso a Internet distintos. Se realiza la instalación de un tercero con el objetivo de tener un punto de reserva por si fallara alguno de ellos.

#### 4.2. Verificación del Despliegue

Una vez llevada a cabo la instalación, cuando los dispositivos así como la luminaria obtienen alimentación a través de la tensión de red, los nodos comienzan a consultar y actualizar los recursos especificados a nivel de aplicación mediante el protocolo CoAP. Dentro de estos recursos destacan algunos como el perfil de iluminación que debe utilizar el driver o el estado del propio driver. La instalación se enciende de forma sincronizada

a través del cuadro eléctrico, donde se encuentra instalado un reloj astronómico que gestiona el encendido y apagado de la red de alumbrado público.

En la Figura 44 se muestra un mapa en la aplicación de STELARIA donde se puede monitorizar el tráfico de la red de comunicaciones. Como se puede apreciar, no todos los nodos envían directamente al nodo concentrador el tráfico sino que utilizan caminos alternativos a través de nodos próximos. De esta forma se valida que la red mallada funciona como se esperaba y se recibe el tráfico de todos los nodos que conforman la red.



Figura 44 Mapa del despliegue en STELARIA. Puntos encendidos enviando tráfico

La consulta o actualización de recursos (operaciones GET y PUT en CoAP) se realiza para cada nodo cada 2 minutos y 10 segundos. Respecto a este tiempo programado pueden existir variaciones de hasta 2 segundos introducidas por el retardo debido a la propagación del mensaje a través de la red mallada y la latencia de la red de telefonía móvil. En la Figura 45 se muestra un ejemplo para uno de los nodos instalados.

```

21:04:51 INFO OperacionRecurso: 44 - Recibido GET en el recurso: /003a0041-30333932-0f473535/op
21:07:00 INFO EstadoRecurso: 56 - Recibido PUT en el recurso: /003a0041-30333932-0f473535/estado
21:09:10 INFO OperacionRecurso: 44 - Recibido GET en el recurso: /003a0041-30333932-0f473535/op
21:11:20 INFO EstadoRecurso: 56 - Recibido PUT en el recurso: /003a0041-30333932-0f473535/estado
21:13:29 INFO OperacionRecurso: 44 - Recibido GET en el recurso: /003a0041-30333932-0f473535/op
21:15:39 INFO EstadoRecurso: 56 - Recibido PUT en el recurso: /003a0041-30333932-0f473535/estado
21:17:49 INFO OperacionRecurso: 44 - Recibido GET en el recurso: /003a0041-30333932-0f473535/op
21:19:59 INFO EstadoRecurso: 56 - Recibido PUT en el recurso: /003a0041-30333932-0f473535/estado
21:22:08 INFO OperacionRecurso: 44 - Recibido GET en el recurso: /003a0041-30333932-0f473535/op
21:24:18 INFO EstadoRecurso: 56 - Recibido PUT en el recurso: /003a0041-30333932-0f473535/estado
21:26:28 INFO HoraSyncRecurso: 52 - Recibido GET en el recurso: /003a0041-30333932-0f473535/time
21:28:37 INFO EstadoRecurso: 56 - Recibido PUT en el recurso: /003a0041-30333932-0f473535/estado
21:30:47 INFO CSMIRecurso: 54 - Recibido GET en el recurso: /003a0041-30333932-0f473535/csmi
21:32:57 INFO EstadoRecurso: 56 - Recibido PUT en el recurso: /003a0041-30333932-0f473535/estado
21:35:06 INFO OperacionRecurso: 44 - Recibido GET en el recurso: /003a0041-30333932-0f473535/op
21:37:16 INFO EstadoRecurso: 56 - Recibido PUT en el recurso: /003a0041-30333932-0f473535/estado
21:39:26 INFO OperacionRecurso: 44 - Recibido GET en el recurso: /003a0041-30333932-0f473535/op
21:41:36 INFO EstadoRecurso: 56 - Recibido PUT en el recurso: /003a0041-30333932-0f473535/estado
    
```

Figura 45 Frontal CoAP. Filtrado por MAC de nodo. Consulta de recursos

La captura se ha realizado desde el servidor remoto que implementa la capa de aplicación CoAP (también denominado frontal CoAP) y mantiene los recursos expuestos para cada dispositivo. Como se puede apreciar, este tiempo de 2 minutos y 10 segundos se mantiene a lo largo de las consultas y actualizaciones de los recursos realizadas.

Desde la plataforma STELARIA pueden actualizarse los recursos consultados por cada dispositivo. Uno de estos es el perfil de iluminación que debe ejecutar cada driver incorporado en la luminaria, que indica con qué valor de potencia y a qué hora debe regularse el flujo lumínico a través de la regulación de la corriente. Esto es posible dado que los drivers de ELT de la gama iLC pueden configurar la corriente de salida sobre la carga LED y con ello regular el consumo de potencia.

En la Figura 46 se recoge un ejemplo de perfil de iluminación diseñado a través STELARIA. En este caso, es un perfil que proporciona un ahorro de potencia del 45% frente a un perfil que aplique el 100% de la potencia posible durante toda la noche. Dicho perfil se configura para la totalidad de los nodos desplegados.

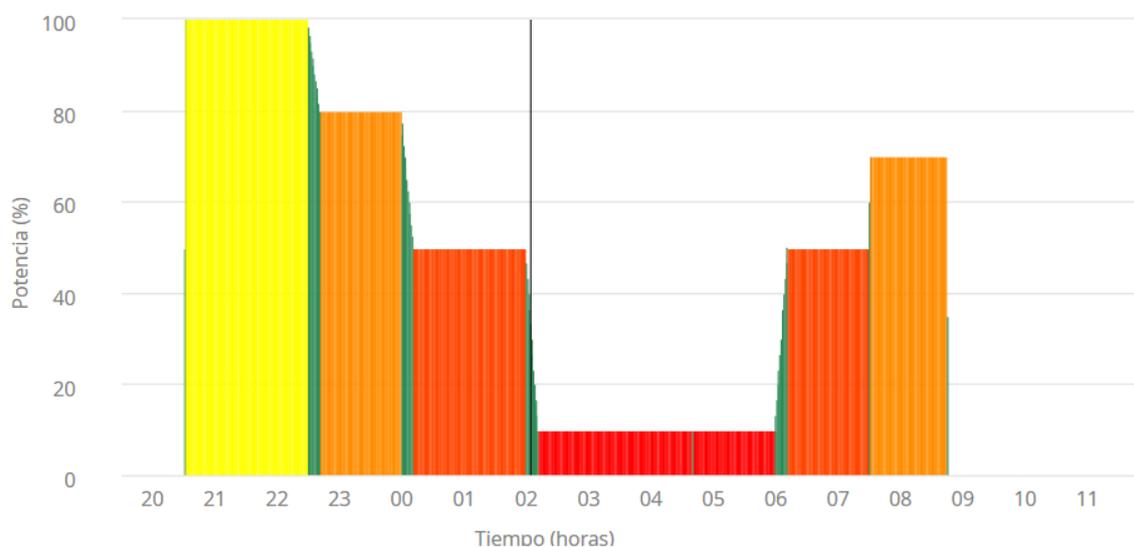


Figura 46 Ejemplo de perfil de iluminación diseñado. 7 niveles y 45% de ahorro

STELARIA permite actualizar un recurso de varios nodos a la vez a través del concepto de grupo lógico. Un grupo lógico permite asociar distintos nodos de forma que el instalador pueda relacionar dicho grupo con una calle o un conjunto de calles, una plaza, un lugar emblemático de la localidad etc.

### 4.3. Beneficios del Despliegue y Estimación de Ahorro

Dentro de los beneficios que se obtiene tras el despliegue de una red de comunicaciones que permite la interacción con dispositivos como una fuente de alimentación en un ámbito como el alumbrado público, destacan los siguientes:

- Gestión eficiente del alumbrado. Ahorro de tiempo en planificación de revisiones y visualización remota del estado de cada punto de luz.

- Control y programación remota de corriente de salida de la fuente de alimentación. Ahorro en consumo energético en base al perfil de iluminación diseñado.
- Detección y aviso de fallos.

A continuación se recoge un estudio basado en el ahorro de consumo energético obtenido tras el despliegue en la instalación descrita. Durante el proceso de instalación se obtuvieron distintas muestras de las luminarias reemplazadas para su evaluación y estudio. Todas ellas se correspondían con luminarias basadas en tecnología VSAP (Vapor de Sodio a Alta Presión). Esta tecnología se encuentra extendida en el mundo del alumbrado, más en concreto en el alumbrado público [21].

Las luminarias reemplazadas disponían, a diferencia de la tecnología LED que utiliza fuentes de alimentación electrónicas, un balasto electromagnético que proporciona una potencia de 150W a una tensión de alimentación de 230V nominales. Sin embargo, gracias a STELARIA, se ha podido apreciar que la tensión medida en la red desplegada es de 242V promedio. Las medidas obtenidas para este valor de tensión de alimentación reportan un consumo de 186 W por luminaria, así como una corriente consumida entorno a 1 A y un factor de potencia de 0.8. La Figura 47 muestra un ejemplo de las medidas realizadas sobre uno de los equipos reemplazados.



Figura 47 Ejemplo de medida en equipo reemplazado.

En esta instalación, los driver instalados de ELT proporcionan una potencia de 60W teniendo en cuenta el factor de eficiencia del driver (0.97). En la Tabla 5 se muestran recogidas las principales características de la instalación.

Característica	Valor
Tipo de Tecnología Reemplazada	VSAP
Consumo (W)	186
Tipo de Tecnología Instalada	LED
Driver	ILC 75W
Potencia Máxima (W)	75
Tensión Carga LED (V)	110
Corriente de Salida Programada (mA)	530
Potencia Consumida (W)	60
Duración Promedio Anual de la Noche (horas/día)	11,7
Precio Factura Estándar kWh (€)	0,123

Tabla 5 Tabla resumen de la instalación

En base a los datos obtenidos se puede obtener una estimación de ahorro extrapolando los ahorros obtenidos para un punto de luz a la totalidad de la instalación. En este caso se utiliza el precio por kWh obtenido de la media de tarifas disponibles de contratación, 0,123 €. Para las estimaciones de ahorro se ha calculado la duración promedio de la noche como el promedio entre la duración de la noche más larga y la más corta del año. En este caso la duración es de 11,7 horas. La Tabla 6 muestra una comparativa del coste por consumo energético entre la luminaria reemplazada y la instalada.

Luminaria Antigua		Luminaria Nueva	
Consumo (kWh/día)	2,1762	Consumo (kWh/día)	0,702
Precio (€)	0,267	Precio (€)	0,086

Tabla 6 Tabla comparativa de coste de consumo energético entre luminarias

Existe una reducción de 0,267 € diarios a 0,086 € diarios, es decir, una reducción del 67,74%. Sin embargo, a este ahorro hay que añadir el ahorro obtenido por el perfil de iluminación gestionado y programado a través de STELARIA y la red de comunicaciones diseñada (ver Figura 46). Dicho perfil proporciona un ahorro del 45% adicional, por lo que la comparativa, teniendo en cuenta la incorporación de los dispositivos diseñados, quede tal y como se muestra en la Tabla 7.

Luminaria Antigua		Luminaria Nueva con STELARIA	
Consumo (kWh/día)	2,1762	Consumo (kWh/día)	0,386
Precio (€)	0,267	Precio (€)	0,047

Tabla 7 Tabla comparativa de coste de consumo incluyendo STELARIA

Por lo tanto, el coste promedio sería de 0,047 € diarios por punto de luz frente a los 0,267 de la luminaria reemplazada. Esto supone un ahorro definitivo del 82,3% diario, un 14,5% más que sin el sistema STELARIA. Extrapolando dichas cifras de consumo energético a la totalidad de los 41 puntos que conforman el despliegue, se obtienen los resultados de ahorro mostrados en la Tabla 8.

Tabla Resumen	
<b>Ahorro diario (€)</b>	9,02 €
<b>Ahorro mensual (€)</b>	270,60 €
<b>Ahorro anual (€)</b>	<b>3.292,27 €</b>

Tabla 8 Tabla de ahorro económico para la instalación

## 5. Conclusiones y Líneas de Trabajo Futuro

---

### 5.1. Conclusiones

Tras la finalización de este trabajo fin de máster y analizando sus correspondientes resultados se observa que se han alcanzado los distintos objetivos:

- Diseño de una red de comunicaciones basada en RF. Estudio y selección de soporte tecnológico y capa de protocolos a nivel de red, transporte y aplicación.
- Estudio de sistema operativo Contiki OS. Análisis de herramientas disponibles, emulación de red de nodos para posterior validación de la pila de protocolos seleccionada.
- Estudio y selección de familia de dispositivos embebidos. Justificación y selección de familia de microcontroladores ARM de 32 bits para implementación del sistema operativo Contiki OS sobre éste.
- Diseño electrónico, fabricación y diseño de procesos de test para fuente de alimentación, placa de conectividad 2G/3G y placa de RF mediante microcontrolador y transceptor radio en banda ISM.
- Validación del diseño electrónico a través de pruebas de comunicación entre dispositivos mediante el transceptor radio.
- Implementación de nivel de aplicación CoAP para conexión a través de distintos puertos de comunicación con sensores, dispositivos digitales, etc.
- Fabricación de pre-serie de PCBs y despliegue de validación sobre instalaciones de ELT. Verificación y validación del diseño firmware y hardware realizado.
- Despliegue de la red de comunicación diseñada sobre proyecto en instalación real con la empresa ELT. Verificación del despliegue. Estudio y análisis de beneficios obtenidos a través de la red diseñada en este trabajo fin de máster.

### 5.2. Líneas de Trabajo Futuro

Como líneas futuras de investigación quedan abiertos distintos puntos de interés enfocados principalmente en:

- Estudio e implementación de otros protocolos de aplicación distintos a CoAP como MQTT.
- Estudio e implementación de la selección de protocolos realizada sobre otros soportes físico tales como PLC o RF a través del estándar Wireless M-BUS.
- Estudio y programación de firmware basado en la conexión a sensores de iluminación o presencia para ajuste lumínico en base a nivel de luz o detección y conteo de personas.
- Diseño electrónico y programación de nuevo dispositivo concentrador con acceso a Internet a través de Ethernet.
- Diseño electrónico y programación de dispositivos basados en tecnología NB-IoT (Narrow Band IoT) a través de la red 4G.

# ANEXOS



## ANEXO A. Estándar IEEE 802.15.4

---

El estándar IEEE 802.15.4 define una topología de red tipo LR-WPAN (Low Rate Wireless Personal Area Network) para dispositivos de bajo coste con requisitos de consumo energético y de ancho de banda reducidos. El objetivo principal del estándar mencionado es ofrecer una red de instalación sencilla, que ofrezca flexibilidad de cara a implementar distintos protocolos de aplicación.

A continuación se recogen algunas de las prestaciones que ofrece una red basada en el estándar IEEE 802.15.4:

- Topología de red en estrella o peer-to-peer
- Direcciones únicas de 64 o 16 bits
- Control de acceso al medio basado en CSMA-CA (Carrier Sense Multiple Acces with Collision Avoidance)
- Mensajes con garantía de recepción mediante ACK (Acknowledge)
- Detección de energía
- Indicador de calidad del enlace (LQI, Link Quality Indicator)

De esta forma, quedaría definido tras la selección del estándar los niveles físico y de enlace correspondientes a la pila OSI.

El estándar realiza una diferenciación entre dos tipos de dispositivos que compondrán la red de comunicaciones. Dispositivos de funcionalidad completa, FFD (Full-Function Device) y dispositivos con funcionalidad reducida, RFD (Reduced-Function Device). Un dispositivo FFD podrá actuar de coordinador de la red o como un dispositivo normal. Puede comunicarse con cualquier otro dispositivo y de forma habitual dispone de alimentación externa.

Por otro lado, un dispositivo RFD está pensado para trabajar con requerimientos y recursos limitados, pudiendo comunicarse únicamente con otros FFD's y sin capacidad de encaminar tramas hacia otros dispositivos. De forma habitual trabajan con baterías y no pueden actuar como coordinadores de la red.

El estándar proporciona dos topologías de red (ver Figura 48). Una topología basada en estrella y otra orientada a una comunicación entre dispositivos (P2P). Cada dispositivo dispone de un identificador único de 64 bits, que permitirá identificar al nodo a nivel de enlace (dirección MAC), aunque bajo ciertas condiciones cabe la posibilidad de trabajar con direcciones de 16 bits para reducir el overhead.

El estándar IEEE 802.15.4 no define un nivel de red. Es decir, no define un conjunto de tramas de encaminamiento, sino que en el diseño de la red se debe definir, como se hará a continuación, un protocolo de red.

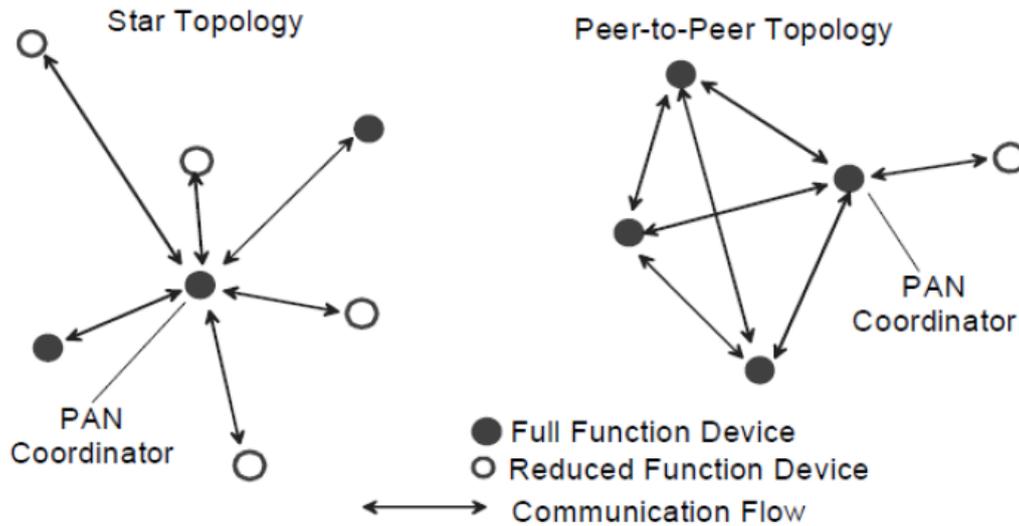


Figura 48 Topologías de red ofrecidas por el estándar IEEE 802.15.4

En referencia al nivel físico, el estándar define un conjunto de modulaciones y tasas de transmisión que se recogen en la Figura 49. En base a los criterios de alcance y cobertura mencionados en el capítulo 2, la banda frecuencial a emplear será la de 868 MHz. Consiste en un único canal entre 868.0 y 868.6 MHz. En dicho canal se emplea una tasa de transmisión de datos de 20 kbps o 40 kbps basada en una modulación BPSK (Binary Phase Shift Keying).

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters		
		Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
780	779–787	1000	O-QPSK	250	62.5	16-ary orthogonal
780	779–787	1000	MPSK	250	62.5	16-ary orthogonal
868/915	868–868.6	300	BPSK	20	20	Binary
	902–928	600	BPSK	40	40	Binary
868/915 (optional)	868–868.6	400	ASK	250	12.5	20-bit PSSS
	902–928	1600	ASK	250	50	5-bit PSSS
868/915 (optional)	868–868.6	400	O-QPSK	100	25	16-ary orthogonal
	902–928	1000	O-QPSK	250	62.5	16-ary orthogonal
950	950–956	—	GFSK	100	100	Binary
950	950–956	300	BPSK	20	20	Binary
2450 DSSS	2400–2483.5	2000	O-QPSK	250	62.5	16-ary orthogonal
UWB sub-gigahertz (optional)	250–750	As defined in 14.4.1				
2450 CSS (optional)	2400–2483.5	As defined in 13.2		250	167 (as defined in 13.4.2)	
		As defined in 13.2		1000	167 (as defined in 13.4.2)	
UWB low band (optional)	3244–4742	As defined in 14.4.1				
UWB high band (optional)	5944–10 234	As defined in 14.4.1				

Figura 49 Estándar IEEE 802.15.4. Frecuencias, modulaciones y tasas disponibles

La capa MAC del estándar IEEE 802.15.4 emplea CSMA-CA (Carrier Sense Multiple Acces with Collision Avoidance). El canal de comunicaciones (espacio radioeléctrico en este caso) será compartido por todos los nodos que conforman la red. Existen dos formas de controlar el acceso al medio.

En primer lugar, los dispositivos esperarán un tiempo aleatorio antes de realizar un envío de mensaje y monitorizarán durante un breve periodo el canal antes de enviar para saber si hay presente una transmisión y no colisionar así con la transmisión del dispositivo vecino.

Del mismo modo, cabe la posibilidad de utilizar una configuración basada en slots temporales (modo beacon-enabled) pensada principalmente para topología en estrella donde se requiere un control temporal de acceso al medio.

Por último, cabe destacar el formato de la trama a nivel de bytes mostrado en la Figura 50. El tamaño de trama disponible a nivel de enlace (L2), es de 127 bytes, denominándose el conjunto de bytes a este nivel MPDU que posteriormente es encapsulado en una trama física denominada PPDU (Physical Protocol Data Unit).

Por lo tanto esto limitará el tamaño y overhead de los niveles posteriores como el nivel de red, nivel de transporte y nivel aplicación, siendo necesario adaptar el tamaño de los mensajes al nivel de enlace mencionado, realizando la fragmentación de los paquetes si fuera necesario.

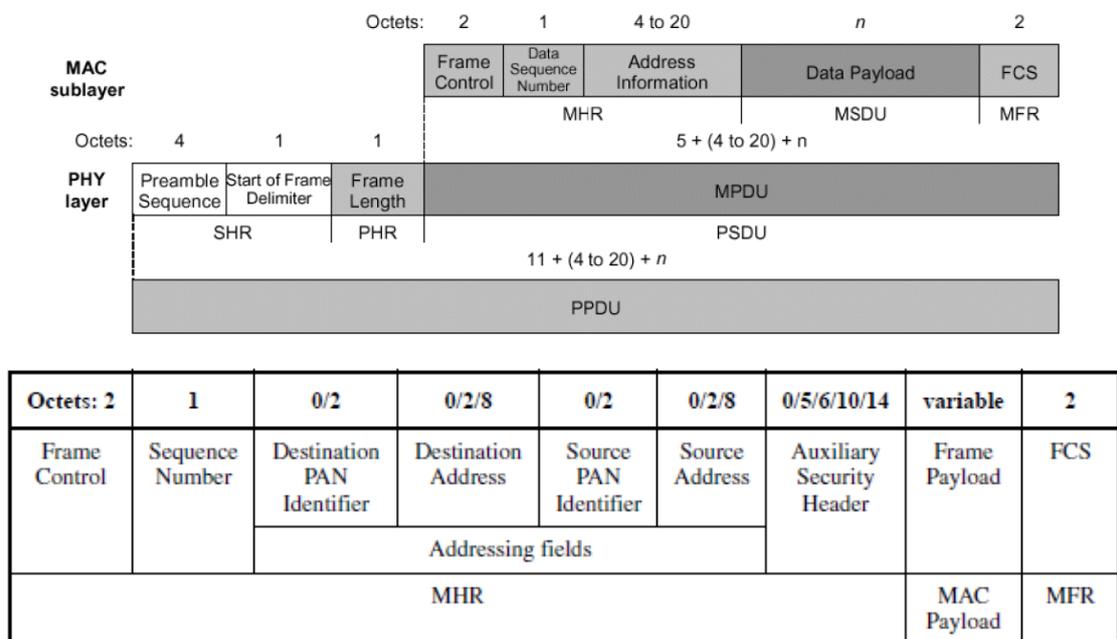


Figura 50 Trama IEEE 802.15.4. PPDU y MPDU

La potencia de transmisión para este tipo de dispositivos está regulada por el ETSI (European Telecommunications Standards Institute) a nivel europeo bajo la normativa EN 300 220-1. En dicha banda frecuencial se permite un máximo de potencia radiada de 25 mW o lo que es lo mismo, 14dBm (ver Figura 51).

<b>Frequency Bands/frequencies</b>	<b>Applications</b>	<b>Maximum radiated power, e.r.p. / power spectral density</b>	<b>Channel spacing</b>
868,000 MHz to 868,600 MHz (see note 4)	Non-specific use	25 mW	No requirement (see note 6)

Figura 51 Cuadro de máxima potencia radiada permitida. Normativa EN 300 220-1

## ANEXO B. Bluetooth Low Energy

Bluetooth Low Energy (BLE) se ha convertido en uno de los protocolos referentes en comunicaciones tras la publicación de su perfil mesh, que permite interconectar dispositivos con tecnología Bluetooth y mejorar los alcances a través de la conformación de una red mallada.

A continuación se recoge un resumen del artículo publicado por el autor de este proyecto final de máster en referencia a la tecnología BLE en la revista Smart-Lighting y en la página web de la empresa ELT.

La pila de protocolos para Bluetooth Low Energy sigue la estructura definida en la Figura 52.

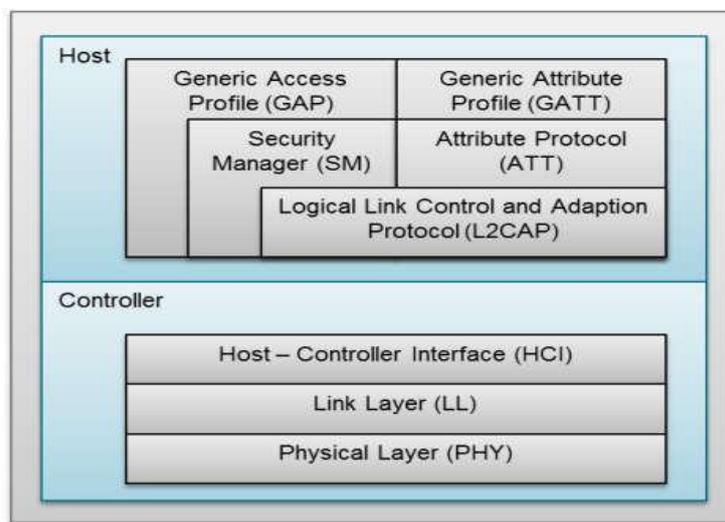


Figura 52 Pila de protocolos de BLE

La capa física contiene la circuitería de comunicaciones capaz de realizar los procesos de modulación y demodulación de señales analógicas y posteriormente transformarlas en símbolos digitales. La tecnología BLE es capaz de utilizar hasta 40 canales de 2 MHz en la banda ISM de 2.4 GHz (ver Figura 53). El estándar emplea la técnica “frequency hopping o saltos en frecuencia”, siguiendo una secuencia de saltos entre los canales frecuenciales mencionados que ofrece un alto grado de robustez frente a interferencias.

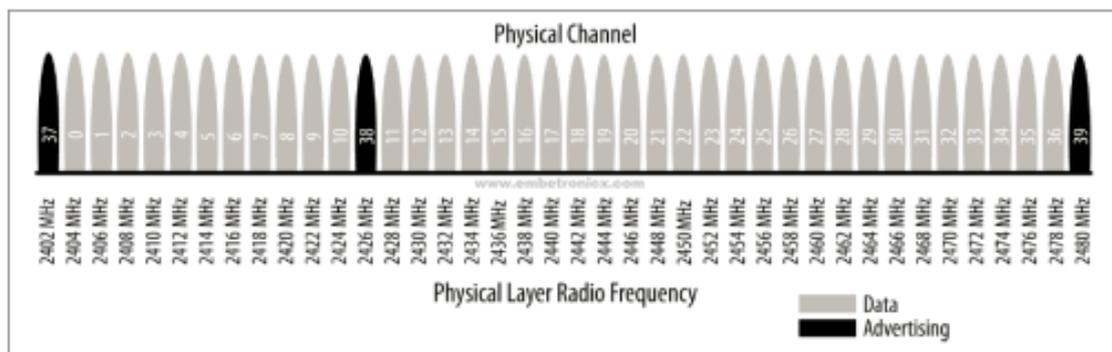


Figura 53 Canales frecuenciales disponibles para BLE. Canales de datos y advertising

La capa de enlace (link layer), se encarga de gestionar características como los requerimientos temporales del estándar, chequeo de mensajes y reenvío de mensajes erróneos recibidos, gestión, filtrado de direcciones etc. Además ofrece la definición de roles (Advertiser, Scanner, Master and Slave) que permiten identificar de forma lógica el rol de cada dispositivo en el proceso de comunicación. El nivel LL es del mismo modo responsable de procesos de control como el cambio de parámetros de la conexión o la encriptación.

HCI (Host Controller Interface) es un protocolo estándar que permite que la comunicación entre un host y un controlador se lleve a cabo a través de un interfaz serie. A modo de ejemplo, en la mayoría de smartphones u ordenadores el host y la aplicación corren en la CPU principal mientras que el controlador está situado en hardware específico y separado, conectado mediante UART o USB. El estándar Bluetooth define HCI como el conjunto de comandos y eventos para la interacción de ambas partes (host y controlador).

La capa L2CAP (Logic Link Control and Adaptation Protocol), se responsabiliza de dos tareas fundamentales en un proceso de comunicación. En primer lugar, el proceso de multiplexación, es decir, la capacidad de dar formato a mensajes provenientes de las capas OSI superiores y encapsularlos en paquetes estándar BLE así como el proceso inverso.

Por otro lado, la fragmentación y recombinación. Paquetes que en nivel de aplicación suponen datagramas de gran cantidad de bytes son fragmentados correctamente adecuándose al MTU de BLE (27 bytes de payload máximo).

Para BLE, la capa L2CAP es la encargada de dar acceso y soporte a los dos protocolos fundamentales. Por un lado, ATT (Attribute Protocol), un protocolo basado en atributos presentados por dispositivo, con arquitectura cliente-servidor, que permite el intercambio de información. Por otro lado, SMP (Security Manager Protocol), protocolo que proporciona un framework para generar y distribuir claves de seguridad entre dos dispositivos.

En el nivel más alto de la capa de protocolos, encontraremos de forma paralela las capas GAP y GATT. Esta primera, GAP (Generic Acces Profile), permite que un dispositivo sea visible hacia el resto de dispositivos y además determina como puede interactuar un dispositivo entre otro. Establece distintas normas y conceptos para estandarizar las operaciones de más bajo nivel como:

- Roles de interacción
- Modos de operación y transición entre ellos
- Procedimientos para establecimiento de comunicación
- Modos de seguridad y procedimientos

Al otro lado, GATT (Generic Attribute Profile), que define como dos dispositivos BLE transfieren información. Este proceso tiene lugar cuando dos dispositivos han superado la fase de establecimiento de comunicación (controlada por GAP) y comienza la transferencia de información pudiendo ser de forma bidireccional.

### **BLE en el Sector de la Iluminación**

Como era de esperar, el sector de la iluminación empieza a mostrar su interés en la tecnología Bluetooth Low Energy pensando en sistemas de comunicación para sensorización de ambientes interiores en hogares, oficinas o naves industriales.

En el mundo del alumbrado público, son los sectores de mantenedores de instalaciones y fabricantes quienes encuentran el atractivo a esta tecnología con el fin de optimizar sus recursos y encontrar de una forma poco costosa un canal de información y actuación sobre cada punto de luz.

Sin embargo, a diferencia de otros modelos de comunicación, en BLE no existe una estandarización a nivel de aplicación. Es decir, los servidores y los recursos/atributos que estos exponen no quedan definidos en un documento y bajo un conjunto de reglas estándar. Por ello, existen en el mercado multitud de soluciones Bluetooth y algunas BLE, pero no existe ese grado de interoperabilidad entre productos de distintos fabricantes. En otras palabras, podemos afirmar que existe un soporte estándar para una comunicación e intercambio de información no estándar.

REF: <https://smart-lighting.es/bluetooth-low-energy/>

REF: <http://www.elt.es/en/blog/bluetooth-low-energy-ble-introduction-technology>

## ANEXO C. Algoritmo RPL

En el presente Anexo se recoge un ejemplo del algoritmo RPL para un encaminamiento de 5 dispositivos y un concentrador con acceso a Internet. La métrica empleada es el ETX, la OF es minimizar el ETX y el rango se calcula sumando el ETX monitorizado al rango del nodo candidato a ser padre (esta función OF se denomina OF Zero, RFC 6552). Los valores mostrados corresponden al último ETX monitorizado (ver Figura 54).

Los nodos A y C se comunicarán con el Edge Router (ER) de forma directa con rangos 1,2 y 1,4 respectivamente. Cuando C reciba el mensaje RPL de A que le indica que dispone de un rango de 1,2 y un ETX de 1,8, C encontrará un camino hacia el ER a través de A con un rango  $1,8 + 1,2 = 3$ . Dado que este valor es mayor que el 1,4 que obtiene a través del enlace directo, mantendrá su comunicación con el ER de forma directa.

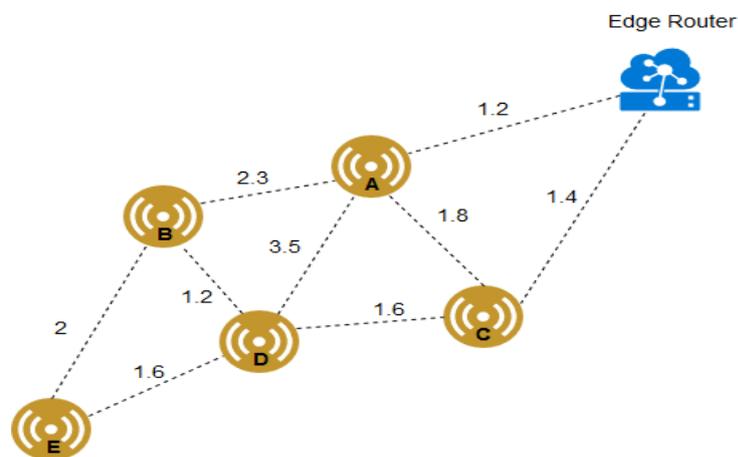


Figura 54 Enlaces disponibles durante ejecución del algoritmo RPL

Cuando A y C propaguen a través del protocolo RPL sus rangos, serán recibidos por B y D. B encontrará un camino a través de A con un rango de  $2,3 + \text{RangoA} = 2,3 + 1,2 = 3,5$  y D encontrará un camino a través de A con rango 4.7 o a través de C con rango 3, por lo que seleccionará C como padre de reenvío. Esto se realiza de forma consecutiva de modo que los caminos seleccionados y el rango disponible será el mostrado en la Figura 55.

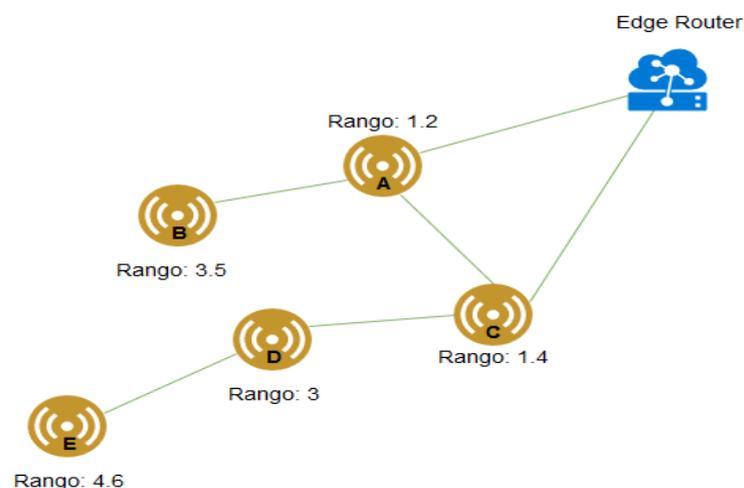


Figura 55 Rutas de encaminamiento seleccionado tras ejecución de RPL

## ANEXO D. Sistema Operativo Contiki OS

En este anexo se recoge información referente al estudio, análisis y modificaciones realizadas sobre el RTOS Contiki OS.

### Procesos en Contiki OS

Un proceso se puede ejecutar bajo cualquiera de los dos contextos de ejecución que existen: cooperativo o con prioridad (ver Figura 57). El código cooperativo corre secuencialmente respecto a otro código cooperativo. El código con prioridad, dispone de capacidad para parar la ejecución de un proceso cooperativo y tomar la CPU hasta su finalización. Los procesos de Contiki OS corren en modo cooperativo, mientras que las interrupciones utilizan contexto preferente. Cuando el código preferente obliga a parar al cooperativo, éste no volverá a tomar CPU hasta que termine el código con prioridad.

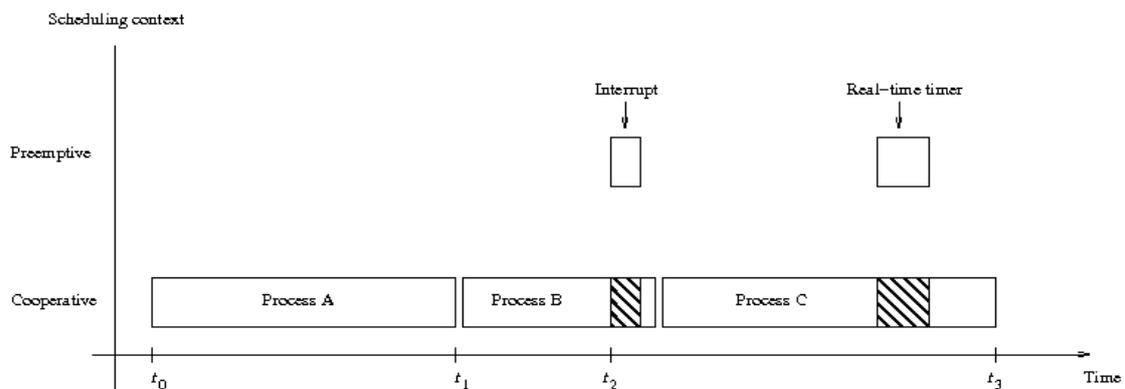


Figura 56 Ejecución de procesos en Contiki OS. Preemptive y cooperative

Un proceso de Contiki, está conformado por dos partes claramente diferenciadas. En primer lugar, el bloque de control almacenado en la RAM que contiene información como el nombre del proceso, estado del proceso y el puntero al hilo del ejecución del proceso.

Por otro lado, disponemos del hilo de ejecución del proceso, correspondiente al código del proceso almacenado en la ROM que se ejecutará cuando se produzca el evento que dispara a dicho proceso. En la Figura 57 se muestra un ejemplo básico de ambas partes mencionadas. El bloque de control requiere únicamente 2 bytes de memoria, mientras que el hilo de ejecución está basado en el concepto de protohilo.

Un protohilo es una forma de estructurar el código de que permite al sistema hacer correr otras funcionalidades mientras el código está esperando a que se produzca un evento. No es algo propio de Contiki OS sino que es utilizado por otros sistemas operativos. El funcionamiento es muy similar al de un hilo de ejecución de un sistema operativo tradicional salvo que el overhead de memoria es muy inferior (reduce las restricciones de memoria).

```
PROCESS(hello_world_process, "Hello world process");
```

```
PROCESS_THREAD(hello_world_process, ev, data)
{
    PROCESS_BEGIN();

    printf("Hello, world\n");

    PROCESS_END();
}
```

Figura 57 Ejemplo de bloque de control e hilo de ejecución para un proceso

### Análisis de Trama en Emulación de Contiki OS

A continuación se recoge un estudio de trama capturado mediante la herramienta Cooja de emulación que ofrece el sistema operativo Contiki OS. Se ha capturado el tráfico de una red de nodos emulados. El objetivo de esta captura durante la emulación es identificar las cabeceras de los distintos niveles de comunicación que se encuentran en una red 6LoWPAN, como son el nivel MAC (marcado por el estándar IEEE 802.15.4), cabeceras IP, UDP o protocolos de aplicación que se encuentre por encima.

Con la herramienta de emulación Cooja se obtiene un fichero de extensión .pcap, donde se recogen los mensajes intercambiados a través del espacio radioeléctrica de forma emulada. El tamaño de los paquetes muestra variabilidad en cuánto al tamaño se refiere debido a la existencia de distinto tipo de mensajes como pueden ser mensajes de enrutamiento RPL sobre ICMPv6 o mensajes puramente UDP de transmisión de datos.

En este caso, se recoge la captura de un mensaje de tipo UDP intercambiado entre el nodo 3 y nodo 7. En la Figura 58 se encuentra el esquema que sigue una cabecera a nivel MAC para el estándar IEEE 802.15.4. El objetivo es identificar cuál es la relación con los datos proporcionados por la captura a nivel MAC, proporcionados en la parte inferior de la imagen.

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

Figura 58 Cabecera Nivel MAC estándar IEEE 802.15.4

**IEEE 802.15.4 Data, Dst: NitLab\_03:00:03:03:03, Src: NitLab\_07:00:07:07:07**  
 Frame Control Field: 0xdc61, Frame Type: Data, Acknowledge Request, PAN ID Compression, Destination Addressing Mode: Long/64-bit, Frame Version: IEEE Std 802.15.4-2006, Source Addressing Mode: Long/64-bit  
 Sequence Number: 20  
 Destination PAN: 0xabcd  
 Destination: NitLab\_03:00:03:03:03 (00:12:74:03:00:03:03:03)  
 Extended Source: NitLab\_07:00:07:07:07 (00:12:74:07:00:07:07:07)  
 FCS: 0x0561 (Correct)

Se puede observar la presencia de todos los campos que proporciona el estándar que iremos comentando. En primer lugar, el Frame Control, que contiene una serie de campos como el Frame Type, Acknowledge Request (indica si es necesario un ACK como respuesta), PAN ID Compression (si existe compresión en los campos Destination/Source PANID, y el tipo de compresión aplicada en los Destination/Source addresses. Esto ocupa un total de 2 bytes.

A continuación el número de identificación de trama (Sequence Number) con un 1 byte. En función de los valores indicados en el Frame Control, se obtiene un tamaño de trama u otro. Por ejemplo, en este caso no se aplica compresión de direcciones destino y fuente MAC, por lo que son necesarios 8 bytes para cada una de ellas. Sin embargo, sí existe supresión del PAN ID para la fuente, por lo que solo viajan los 2 bytes del PAN ID destino. En total, estos campos de direccionamiento ocupan  $8+8+2 = 18$  bytes.

Por último, habría que añadir a esto el campo de verificación o Checksum (FCS) con un total de 2 bytes. De esta forma, el total ocupado a nivel de acceso al medio (MAC) por el estándar es de 23 bytes.

Nivel de Red

En este caso entra en funcionamiento la compresión ofrecida por la capa de adaptación 6LoWPAN. El protocolo requiere dicha compresión, ya que a nivel de MTU existe una limitación por trama de 127 bytes, ocupando una cabecera IPv6 tradicional 40 bytes. La sustitución se realiza añadiendo una cabecera principal que ocupa 2-3 bytes en función de si existe contexto o no (utilizando la compresión IPHC, RFC 6282, ver Figura 59). Esta cabecera nos indicará que campos han sido comprimidos, cómo, y de qué tipo es la siguiente cabecera que nos vamos a encontrar.

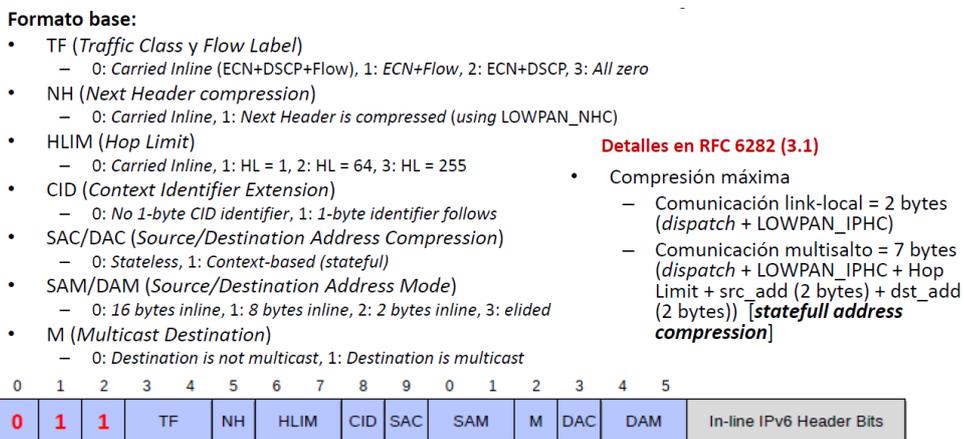


Figura 59 Cabecera IPHC en 6LoWPAN

En esta figura se encuentra un resumen de los campos que dispone la cabecera IPHC. Los campos TF, NH y HLIM, corresponden a los campos TrafficClass y Flow Label, Next Header y Hop Limit de IPv6 tradicional, cuya cabecera se muestra en la parte inferior. Los campos SAC SAM DAC y DAM indican que tipo de compresión se aplica en el direccionamiento y como han sido calculadas.

#### IPHC Header

```

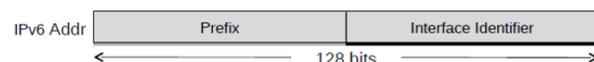
011. .... = Pattern: IP header compression (0x03)
...1 1... .... = Traffic class and flow label: Version, traffic
class, and flow label compressed (0x3)
.... .0.. .... = Next header: Inline
.... ..10 .... = Hop limit: 64 (0x2)
.... .... 1... = Context identifier extension: True
.... .... .1.. = Source address compression: Stateful
.... .... ..11 = Source address mode: Compressed (0x0003)
.... .... .... 0... = Multicast address compression: False
.... .... .... .1.. = Destination address compression: Stateful
.... .... .... ..10 = Destination address mode: 16-bits inline (0x0002)
0000 .... = Source context identifier: 0x0
.... 0000 = Destination context identifier: 0x0

```

Se muestran los datos capturados con WireShark. Éste indica el campo TF = 3, lo que supone que no exista información de Traffic Class ni de FlowLabel, NH = 0, por lo que vendrá a continuación sin compresión y el HopLimit se define a 64, tomando el campo un valor HL = 2. Del mismo modo, se indica que la dirección IPv6 de la fuente se calcula mediante contexto y está comprimida (no se envía) ya que el campo SAM = 3, indicando que la IP se obtiene a partir del nivel de enlace MAC. Las dirección destino sin embargo utiliza compresión a modo de 16 bits. Los posibles modos de compresión se muestran a continuación en la Figura 60.

- Compresión de IPv6 *unicast*

M = 0: Destination address is not Multicast



- Direcciones fuente y destino: 2 modos

128 bits	SAM/DAM = 00	IPv6 Address Bits [0,127]	inline	:: (SAC) – reserved (DAC)
64 bits	SAM/DAM = 01	IPv6 Address Bits [64,127]	[64-bit prefix elided] FE80:: 64 + IID (64)</td <td>P(CID)/64 + IID (64)</td>	P(CID)/64 + IID (64)
16 bits	SAM/DAM = 10	Bits [112,127]	[112-bit prefix elided] FE80::00FF:FE00:: 112 + IID (16)</td <td>[P/64+ IID(48)](CID) + IID (16)</td>	[P/64+ IID(48)](CID) + IID (16)
	SAM/DAM = 11	[Full 128-bit address elided, IID derived from link-layer]	FE80:: 64 + IID-EUI-64 (64)</td <td>[P+IID] (X) + IID-EUI-64 (128-X)</td>	[P+IID] (X) + IID-EUI-64 (128-X)

- Modo *stateless* (SAC/DAC = 0)
  - El prefijo es el local de enlace FE80::
- Modo *stateful* - basado en contexto (SAC/DAC = 1)
  - El prefijo se toma del contexto almacenado (hasta 16)
    - » CID = 0, utiliza ContextID = 0
    - » CID = 1, incluye el ContextID de 4 bits para fuente y destino

Figura 60 Posibles niveles de compresión en dirección IPv6 en 6LoWPAN

Dado que el campo NH indica que las siguientes cabeceras vienen inline, la siguiente cabecera que encontramos es HopbyHop Option Header ( LOWPAN\_NHC), que ocupa 1 byte. Como vienen inline, el orden de las cabeceras sigue el orden de IPv6. La cabecera de IPv6 hop by hop tiene un tamaño de 8 bytes donde muestra distintos parámetros como NextHeader, que sería UDP, Longitud en bytes, RPL options, con un total de 8 bytes.

#### IPv6 Hop-by-Hop Option

```

Next Header: UDP (17)
Length: 0
[Length: 8 bytes]
RPL Option
  Type: RPL Option (0x63)
  Length: 4
  Flag: 0x00
  RPLInstanceID: 0x1e
  Sender Rank: 0x0357

```

A nivel de red (capa 3) encontramos un total de 2 Bytes (IPHC) + 1 Byte (contexto) + 2 Bytes (bytes dirección destino) + 1 Byte (cabecera HopbyHop) + 8 Bytes (HopbyHop) = 14 Bytes.

#### Nivel de Transporte y Aplicación

A nivel de transporte se emplea UDP con una cabecera de 8 bytes, y un payload de 23 bytes.

#### User Datagram Protocol, Src Port: 8765, Dst Port: 5678

```

Source Port: 8765
Destination Port: 5678
Length: 31
Checksum: 0xd6c0 [unverified]
[Checksum Status: Unverified]
[Stream index: 0]

```

#### Data (23 bytes)

```

Data: 48656c6c6f20362066726f6d2074686520636c69656e74
Text: Hello 6 from the client
[Length: 23]

```

## Estudio y Modificaciones sobre Contiki OS

A continuación se recoge un estudio, análisis y modificaciones llevadas a cabo sobre la implementación de la pila TCP/IP y el algoritmo RPL en Contiki OS.

### Análisis del algoritmo RPL

En primer lugar se encuentra la función *handle\_periodic\_timer* (ver Figura 61) encargada de gestionar el DODAG mediante RPL, purgando rutas y recalculando rangos. Para ello se utiliza la estructura *timer\_periodic\_timer* de tipo callback timer. En caso de que se haya superado el *RPL\_DIS\_INTERVAL*, se envía un *dis\_output*. Es decir, el *RPL\_DIS\_INTERVAL* gestiona la frecuencia de envío del DIS (DODAG Information Solicitation) y viene con valor por defecto 60 segundos.

```

/*-----*/
static void
handle_periodic_timer(void *ptr)
{
    rpl_purge_routes();
    rpl_recalculate_ranks();

    /* handle DIS */
    #if RPL_DIS_SEND
        next_dis++;
        if(rpl_get_any_dag() == NULL && next_dis >= RPL_DIS_INTERVAL) {
            next_dis = 0;
            dis_output(NULL);
        }
    #endif
    ctimer_reset(&periodic_timer);
}
/*-----*/

```

Figura 61 Funcion handler periodic timer. Gestión de las rutas RPL

La función *handle\_periodic\_timer* es llamada desde el *rpl\_init* (Figura 62) mediante la función *rpl\_reset\_periodic\_timer*, que realiza un *ctimer\_set* del *periodic\_timer*, y llama a la función *handle\_periodic\_timer*. Cada segundo el código entrada en el *handle\_periodic\_timer* y la variable *next\_dis*, incrementará

```

void
rpl_init(void)
{
    uip_ipaddr_t rplmaddr;
    PRINTF("RPL started\n");
    default_instance = NULL;

    rpl_daod_init();
    rpl_reset_periodic_timer();
    rpl_icmp6_register_handlers();

    /* add rpl multicast address */
    uip_create_linklocal_rplnodes_mcast(&rplmaddr);
    uip_ds6_maddr_add(&rplmaddr);

    #if RPL_CONF_STATS
        memset(&rpl_stats, 0, sizeof(rpl_stats));
    #endif

    RPL_OF.reset(NULL);
}

```

```

/*-----*/
void
rpl_reset_periodic_timer(void)
{
    next_dis = RPL_DIS_INTERVAL / 2 +
        ((uint32_t)RPL_DIS_INTERVAL * (uint32_t)random_rand()) / RANDOM_RAND_MAX -
        RPL_DIS_START_DELAY;
    ctimer_set(&periodic_timer, CLOCK_SECOND, handle_periodic_timer, NULL);
}
/*-----*/

```

Figura 62. Inicialización del algoritmo RPL y procesos Contiki asociados

En referencia a los mensajes DIO (DODAG Information Object), desde la capa de aplicación se llama a la función `rpl_set_root`. Esto solo lo podrá realizar aquel nodo que trabaje como nodo ROOT, de forma que disparará la creación del DODAG mediante el envío del primer DIO.

En `rpl_set_root` al final se termina llamando a `rpl_reset_dio_timer` (ver Figura 63), donde a su vez se llama a la función `new_dio_interval()`.

```

/*-----*/
/* Resets the DIO timer in the instance to its minimal interval. */
void
rpl_reset_dio_timer(rpl_instance_t *instance)
{
#ifdef RPL_LEAF_ONLY
    /* Do not reset if we are already on the minimum interval,
       unless forced to do so. */
    if(instance->dio_intcurrent > instance->dio_intmin) {
        instance->dio_counter = 0;
        instance->dio_intcurrent = instance->dio_intmin;
        new_dio_interval(instance);
    }
#endif
#ifdef RPL_CONF_STATS
    rpl_stats.resets++;
#endif /* RPL_CONF_STATS */
#ifdef RPL_LEAF_ONLY /*
}
/*-----*/

/*-----*/
static void
new_dio_interval(rpl_instance_t *instance)
{
    uint32_t time;
    clock_time_t ticks;

    /* TODO: too small timer intervals for many cases */
    time = 1UL << instance->dio_intcurrent;

    /* Convert from milliseconds to CLOCK_TICKS. */
    ticks = (time * CLOCK_SECOND) / 1000;
    instance->dio_next_delay = ticks;

    /* random number between I/2 and I */
    ticks = ticks / 2 + (ticks / 2 * (uint32_t)random_rand()) / RANDOM_RAND_MAX;

    /*
     * The intervals must be equally long among the nodes for Trickle to
     * operate efficiently. Therefore we need to calculate the delay between
     * the randomized time and the start time of the next interval.
     */
    instance->dio_next_delay -= ticks;
    instance->dio_send = 1;

#ifdef RPL_CONF_STATS
    /* reset the redundancy counter */
    instance->dio_counter = 0;

    /* schedule the timer */
    PRINTF("RPL: Scheduling DIO timer %lu ticks in future (Interval)\n", ticks);
    ctimer_set(&instance->dio_timer, ticks, &handle_dio_timer, instance);
}
/*-----*/

```

Figura 63 Gestión mensajes DIO. Timer asociado y función de control

Cuando el timer configurado anteriormente llegue a su tiempo configurado, se llamará a la función *handle\_dio\_timer* (ver Figura 64), que realiza el envío a través del interfaz radio de un mensaje DIO mediante el *dio\_output(instance, NULL)*.

```

/*-----*/
static void
handle_dio_timer(void *ptr)
{
    rpl_instance_t *instance;

    instance = (rpl_instance_t *)ptr;

    PRINTF("RPL: DIO Timer triggered\n");
    if(!dio_send_ok) {
        if(uiplib_get_link_local(ADDR_PREFERRED) != NULL) {
            dio_send_ok = 1;
        } else {
            PRINTF("RPL: Postponing DIO transmission since link local address is not ok\n");
            ctimer_set(&instance->dio_timer, CLOCK_SECOND, shandle_dio_timer, instance);
            return;
        }
    }

    if(instance->dio_send) {
        /* send DIO if counter is less than desired redundancy */
        if(instance->dio_redundancy != 0 && instance->dio_counter < instance->dio_redundancy)
#ifdef RPL_CONF_STATS
            instance->dio_totssend++;
#endif /* RPL_CONF_STATS */
        dio_output(instance, NULL);
    }
}

```

Figura 64 Función handler para gestión de envío de mensaje DIO de forma periódica

Además, si no se recibe respuesta de un nuevo vecino, el DIO va incrementando su frecuencia de envío como se observa en la siguiente parte de la función *handle\_dio\_timer()*:

```

/* check if we need to double interval */
if(instance->dio_intcurrent < instance->dio_intmin + instance->dio_intdoubl) {
    instance->dio_intcurrent++;
    PRINTF("RPL: DIO Timer interval doubled %d\n", instance->dio_intcurrent);
}
new_dio_interval(instance);

```

Figura 65 Actualización del timer para envío de mensaje DIO

La variable *dio\_intcurrent*, toma al inicializar la construcción del DODAG o reiniciarse el valor 12 proveniente de `#define RPL_DIO_INTERVAL_MIN 12`. Esto indica que el nodo raíz inicializará la llamada al callback cada  $2^{12}(\text{ms}) = 4 \text{ s}$  aprox.

Cada vez que no se reciban nuevas respuestas o cosas que afecten a la construcción del RPL, se irá reduciendo la frecuencia de envío del DIO, de forma que si se hace *instance->dio\_intcurrent++*, lo que hacemos es que la llamada al callback se realizará cada  $2^{\text{dio\_intcurrent}}$ , duplicando cada vez el tiempo respecto a la llamada anterior, hasta un máximo de  $20 = \text{RPL\_DIO\_INTERVAL\_MIN} + \text{RPL\_DIO\_INTERVAL\_DOUBLINGS}$  (8).

Para el proceso del mensaje DAO (DODAG Advertisement Object), se muestra a continuación de forma gráfica el orden de llamada de las funciones para el envío de éste. Cuando se recibe un DIO, se llama a la función *rpl\_schedule\_dao* (ver Figura 66), de manera que se configura el timer correspondiente de tipo callback para el mensaje DAO.

```

/*-----*/
void
rpl_schedule_dao(rpl_instance_t *instance)
{
    schedule_dao(instance, RPL_DAO_LATENCY);
}
/*-----*/

/*-----*/
static void
schedule_dao(rpl_instance_t *instance, clock_time_t latency)
{
    clock_time_t expiration_time;

    if(rpl_get_mode() == RPL_MODE_FEATHER) {
        return;
    }

    expiration_time = etimer_expiration_time(&instance->dao_timer.etimer);

    if(!etimer_expired(&instance->dao_timer.etimer)) {
        PRINTF("RPL: DAO timer already scheduled\n");
    } else {
        if(latency != 0) {
            expiration_time = latency / 2 +
                (random_rand() % (latency));
        } else {
            expiration_time = 0;
        }
        PRINTF("RPL: Scheduling DAO timer %u ticks in the future\n",
            (unsigned)expiration_time);
        ctimer_set(&instance->dao_timer, expiration_time,
            handle_dao_timer, instance);

        set_dao_lifetime_timer(instance);
    }
}
/*-----*/

```

Figura 66 Planificación para mensajes DAO. Disparado tras recepción de DIO

Es el *dao\_output()* quien envía el mensaje DAO. El envío de este se dispara únicamente cuando se recibe un DIO, aunque existe una variable llamada *RPL\_DAO\_LATENCY* que controla el tiempo de configuración del ctimer cuando es configurado tras recibir un DIO. No tiene una periodicidad fija. Se envía en unicast al nodo padre.

## Análisis de los Procesos de la Pila TCP/IP

Todos los procesos que genera este nivel son gestionados de la siguiente forma:

Al inicializar el proceso *tcpip\_process* (ver Figura 67) desde el *main.c* con la función *process\_start*, dicho proceso va inicializando otra serie de capas/protocolos que se encuentra por encima del nivel de red como ICMP o el algoritmo RPL y se queda bloqueado hasta que surge un evento. Este evento se controla con un etimer (timer de eventos) que se inicializa previamente como se indica a continuación:

```

/*-----*/
PROCESS_THREAD(tcpip_process, ev, data)
{
    PROCESS_BEGIN();

    #if UIP_TCP

        tcpip_event = process_alloc_event();
        #if UIP_CONF_ICMP6
            tcpip_icmp6_event = process_alloc_event();
        #endif /* UIP_CONF_ICMP6 */
        etimer_set(&periodic, CLOCK_SECOND / 2);

        uip_init();
        #ifdef UIP_FALLBACK_INTERFACE
            UIP_FALLBACK_INTERFACE.init();
        #endif
        /* initialize RPL if configured for using RPL */
        #if NETSTACK_CONF_WITH_IPV6 && UIP_CONF_IPV6_RPL
            rpl_init();
        #endif /* UIP_CONF_IPV6_RPL */

        while(1) {
            PROCESS_YIELD();
            eventhandler(ev, data);
        }

        PROCESS_END();
    }
/*-----*/

```

Figura 67 Proceso principal para pila TCP/IP. Inicialización y ejecución del proceso

Se observa cómo se llama al *etimer\_set*, configurando el etimer periodic cada  $CLOCK\_SECOND/2 = 500\text{ms}$ . Esto provocará que cada medio segundo se desbloquee el *PROCESS\_YIELD()* y se ejecute el event handler (ver Figura 68). Además de este timer periodic, hay otra serie de eventos que pueden hacer que se desbloquee el proceso:

- PROCESS\_EVENT\_EXITED
- PROCESS\_EVENT\_TIMER
- TCP\_POLL
- UDP\_POLL
- PACKET\_INPUT



```

etimer_set(&periodic, SEND_INTERVAL);
while(1) {
    PROCESS_YIELD();
    if(ev == tcpip_event) {
        tcpip_handler();
    }

    if(etimer_expired(&periodic)) {
        etimer_reset(&periodic);
        ctimer_set(&backoff_timer, SEND_TIME, send_packet, NULL);
    }

#ifdef WITH_COMPOWER
    if (print == 0) {
        powertrace_print("#P");
    }
    if (++print == 3) {
        print = 0;
    }
#endif

}

PROCESS_END();
}

/*-----*/
static void
tcpip_handler(void)
{
    char *str;

    if(uiplib_newdata()) {
        str = uip_appdata;
        str[uip_datalen()] = '\0';
        printf("DATA recv '%s'\n", str);
    }
}
/*-----*/

static void
send_packet(void *ptr)
{
    static int seq_id;
    char buf[MAX_PAYLOAD_LEN];

    seq_id++;
    PRINTF("DATA send to %d 'Hello %d'\n",
           server_ipaddr.u8[sizeof(server_ipaddr.u8) - 1], seq_id);
    sprintf(buf, "Hello %d from the client", seq_id);
    uip_udp_packet_sendto(client_conn, buf, strlen(buf),
                          &server_ipaddr, UIP_HTONS(UDP_SERVER_PORT));
}
/*-----*/

```

Figura 69 Funciones TCP IP Handler tras recepción de mensaje.

```

#define START_INTERVAL      (15 * CLOCK_SECOND)
#define SEND_INTERVAL      (PERIOD * CLOCK_SECOND)
#define SEND_TIME          (random_rand() % (SEND_INTERVAL))

```

Figura 70 Configuración de los clocks para envío de mensaje UDP como cliente

En la Tabla 9 se recogen los procesos de Contiki OS que intervienen en la comunicación tras la implementación de la pila TCP/IP así como la variable de tiempo que llevan asociados cada uno:

Proceso	Variable que controla periodicidad
UDP (APLICACIÓN)	SEND_INTERVAL y SEND_TIME
RPL – ND6	UIP_DS6_PERIOD = CLOCK_SECOND/10 = 100ms
RPL(PERIODIC)	CLOCK_SECOND = 1s
RPL(DIS)	RPL_DIS_INTERVAL. *
RPL(DIO)	RPL_DIO_INTERVAL. **
IP (etimer periodic tcpip.c)	CLOCK_SECOND/2 = 500 ms ***

Tabla 9 Procesos intervinientes en Contiki OS para pila TCP/IP

\*next\_dis = RPL\_DIS\_INTERVAL / 2 + ((uint32\_t)RPL\_DIS\_INTERVAL \* (uint32\_t)random\_rand()) / RANDOM\_RAND\_MAX - RPL\_DIS\_START\_DELAY;  
(rpl\_timers.c, línea 184)

\*\* Parte de  $2^{12}$ ms = 4seg. Se va duplicando si no recibe novedades.

\*\*\*Solo se resetea el *etimer periodic* de la capa tcpip si existen conexiones udp activas.

### Modificación sobre Métrica RPL. ETX Promediado

Con el fin de mejorar la calidad de la red de comunicación, además de la métrica de enlace empleada por Contiki OS basada en el número de saltos, se ha implementado una métrica basada en el número de retransmisiones esperadas de mensaje (ETX). El algoritmo de adaptación de ruta está basado en la función *neighbour\_link\_callback.c*. Las variables que entran en juego con las siguientes:

- RPL\_DAG\_MC\_ETX\_DIVISOR = 256
- ETX\_ALPHA = 90 (alpha)
- ETX\_SCALE = 100 (escala)
- MAX\_LINK\_METRIC = 10
- Recorded\_etx = etx acumulado hasta ahora = rtx
- Packet\_etx = etx del nuevo paquete = ntx

El algoritmo aplicado es:

$$ETX_{ant} \cdot \alpha + ETX_{new} \cdot \left(1 - \frac{\alpha}{\gamma}\right)$$

Con  $\alpha = 0,9$  y  $\gamma = 1$ , por lo que la ecuación de recursividad es:

$$0.9ETX_{ant} + ETX_{new} \cdot 0.1$$

Es decir, pondera más el valor acumulado. Esto es un algoritmo sencillo de media móvil utilizado para controlar las variaciones, evitando que afecten mucho a la ecuación y el sistema se vuelva inestable. Esto se ha implementado para que la red disponga de mayor estabilidad y los cambios de ruta no se realicen frecuentemente.

Sin embargo, si el sistema deja de comunicarse, y empieza a fallar, terminará afectando a la ecuación, y supondrá un cambio importante al irse acumulando. En caso de no recibir ACK, se puede observar en la función *neighbour\_link\_callback()* como  $\text{packet\_etx} = \text{MAX\_LINK\_METRIC} * \text{RPL\_DAG\_MC\_ETX\_DIVISOR} = 2560$ .

Suponiendo una comunicación que ha ido correctamente en todo momento, la métrica del camino sería 0 y por lo tanto, al llamar a la función *calculate\_path\_metric*, se obtendría que la función devolvería  $p\text{-rank} + \text{link\_metric} = p\text{->Rank}$ , dado que la métrica es 0. Sin embargo, si no se obtuviese respuesta del destinatario mediante un ACK,  $\text{packet\_etx} = 2560$ , y  $\text{new\_etx} = 0.9 * 0 + 0.1 * 2560 = 256$ , por lo que el  $\text{path\_metric} = p\text{-rank} + 256$ . Es decir, para el nodo que realiza el procesado, el vecino con dicho *path\_metric* que no envía el ACK, perderá opciones como candidato para ruta por defecto.

## ANEXO E. Código Aplicación Sniffer

A continuación se recoge el código de la herramienta que permite convertir el tráfico capturado a través del transceptor radio y almacenado en el PC a través de puerto serie en un fichero, en otro nuevo fichero con extensión hex dump con capacidad de ser interpretado por herramientas de análisis de tráfico como Wireshark.

```

package tilog2hex;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

/**
 *
 * @author msousa
 */
public class TILog2Hex {

    /**
     * @param args the command line arguments
     */
    static int SIZE_OF_DATE = 6;
    static int SIZE_OF_DATA_LINE = 16;
    public static void main(String[] args) throws IOException {

        BufferedReader br = new BufferedReader(new FileReader("log2Convert"));

        FileWriter logCompleteFile = new FileWriter("logConverted",false);
        BufferedWriter bb = new BufferedWriter(logCompleteFile);

        String line;
        String filtered;
        while( (line=br.readLine())!=null)
        {
            //String lines[] = line.split("|"); // separamos la fecha, los datos y el rssi que no emplearemos
            String fecha = line.substring(0,line.indexOf("|")-5);
            line = line.substring(line.indexOf("|")+1,line.length());
            bb.write(fecha);
            bb.newLine();

            String data = line.substring(1,line.indexOf("|")); // quitamos espacio en blanco
            for (int i = 0; i<((data.length()/3)/SIZE_OF_DATA_LINE)+1; i++)

            {
                bb.write("00"+i+"0 "); // jamás llegamos a un offset de más de 16*8 = 128 > 127 (MPDU)
                int ini = i*SIZE_OF_DATA_LINE*3;
                int fin = (i+1)*SIZE_OF_DATA_LINE*3;
                if(fin>data.length())
                    fin = data.length();

                String dataLine = data.substring(ini,fin);
                bb.write(dataLine);
                bb.newLine();
            }
            bb.newLine();
        }

        br.close();
        bb.close();

        // TODO code application logic here
    }
}

```

## ANEXO F. Estudio y Simulación del Radioenlace

En el presente anexo se recoge el estudio y simulación del radioenlace así como de un estudio de alcance para los dispositivos diseñados. La simulación del radioenlace consiste en un ejercicio académico realizado para explicar durante el diseño los efectos del ruido y las probabilidades de error de bit. El sistema utilizará la banda no licenciada para dispositivos SRD en 868 MHz.

A continuación se recogen las gráficas obtenidas de la simulación del canal. En primer lugar, en la Figura 71 se muestra una imagen que representa la DEP (Densidad Espectral de Potencia) para una modulación GFSK y una tasa de datos de 40 kbps con portadora en 868 MHz. El canal simulado es un canal AWGN (Additive White Gaussian Noise).

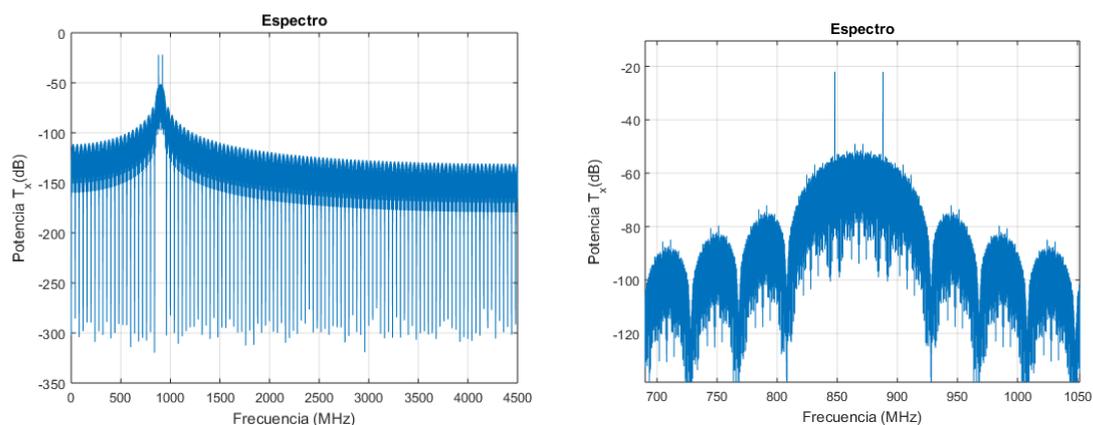


Figura 71 DEP. Comunicación GFSK 40 kbps. Portadora 868 MHz

A continuación, se ha realizado un estudio de BER (Bit Error Rate) en función de la potencia de ruido. Para ello se han simulado 10 casos desde una señal que muestra una SNR (Signal to Noise Ratio) de 1 dB hasta los 10 dB. En la Figura 72 se muestran los resultados de las probabilidades teóricas y las obtenidas en simulación. Para dicha simulación se han utilizado 10E6 símbolos.

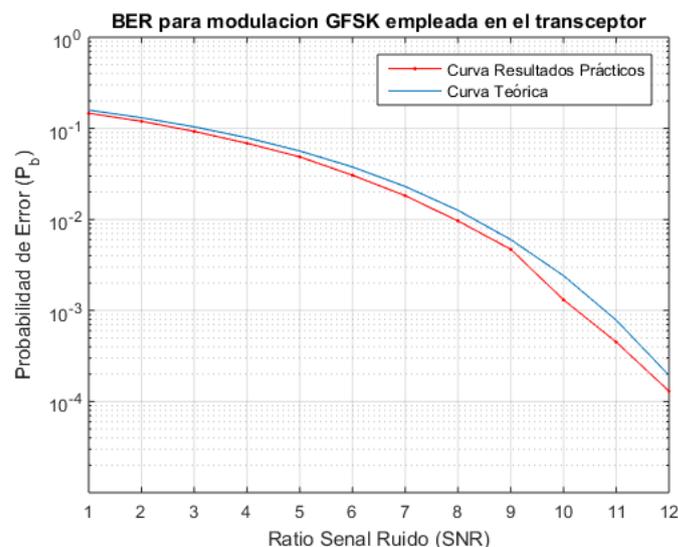


Figura 72 Curvas BER teórica y práctica para canal simulado

A continuación se realiza la estimación de alcance. El sistema trabaja en las bandas de 868 MHz para Europa y 915 MHz para América tal y como queda definido en la normativa que rige el funcionamiento de dicha banda no licenciada. La potencia máxima en bornes de antena transmisora no puede superar los 14 dBm, lo que supone aproximadamente unos 25mW. Atendiendo a esto, se puede obtener una estimación del alcance radio bajo una serie de asunciones y premisas que se deberán justificar correctamente. Para ello, comenzaremos describiendo la ecuación simplificada de balance de canal.

$$P_{Rx} = P_{Tx} + G_s - L_s - L_c - M$$

Siendo:

$P_{Rx}$  : Potencia recibida por el receptor

$P_{Tx}$  : Potencia en bornes de antena del dispositivo transmisor. Máximo en este caso de 14dBm

$G_s$  : Ganancia del sistema. Incluye los efectos de ganancia y Area efectiva

$L_s$  : Pérdidas del Sistema. Incluye las pérdidas por adaptación de etapas, adaptación de impedancias, polaridad etc.

$L_c$ : Pérdidas referentes a la propagación en espacio libre

$M$  : Márgen de Fadding. Margen que se suele reservar tras pruebas empíricas para obtener nivel de señal necesario para correcta demodulación. Se reserva por pérdidas multicamino.

Todos los valores nombrados anteriormente quedan definidos en la ecuación como valores en decibelios (dB)

Partiendo de esto, podemos realizar las siguientes simplificaciones:

$$PIRE = P_{Tx} + G_{Tx} = P_{Tx} + D_{Tx} = P_{Tx} = 14dBm$$

Es decir, se asume que el sistema no dispone de ganancia de transmisión, de modo que la antena es omnidireccional, y por lo tanto las directividades ( $D_{Tx}$  y  $D_{Rx} = 1$ ) (realmente es un dipolo impreso con ganancia 2,15dBi). De esta forma el sistema transmisor queda definido por el PIRE (Potencia Isotrópica Radiada Equivalente) que coincidirá con la potencia transmitida, asumiéndose valor máximo de 14dBm. Se asume del mismo modo que las adaptaciones son idóneas y por lo tanto el factor referente a pérdidas del sistema es nulo,

Por otro lado, la ganancia del sistema tiene en cuenta además el concepto de área efectiva, por lo que:

$$P_{Rx} = P_{Tx} - L_c - M$$

Las pérdidas por propagación en espacio libre quedan definidas como:

$$L_c = \frac{1}{4\pi R^2}$$

Quedaría añadir por último el concepto de área efectiva en recepción. Esto queda definido como

$$A_{eff} = \frac{\lambda^2}{4\pi}$$

Por lo que la ecuación quedará de la siguiente forma:

$$P_{Rx} = P_{Tx} - 20 \log_{10}\left(\frac{\lambda}{4\pi R^2}\right) - M$$

De cara a ofrecer un buen grado de garantía de funcionamiento del sistema, el Margen de Fading que reserva es entre 10 y 20 dB. De esta forma se dispone de todos los valores para despejar la distancia R. Quedaría únicamente identificar cual es la potencia recibida  $P_{Rx}$  que requiere el sistema para funcionar correctamente.

Para ello, se acude al datasheet del transceptor radio que indica lo siguiente:

- RF Performance and Analog Features:
  - High-Performance, Single-Chip Transceiver
    - Excellent Receiver Sensitivity:
      - -123 dBm at 1.2 kbps
      - -109 dBm at 50 kbps

Es decir, hará referencia al concepto de sensibilidad del dispositivo, cuanto potencia hace falta al menos en bornes de antena de recepción para que el sistema funcione correctamente. Siempre vendrán varios valores en función del ancho de banda que consume nuestro sistema. Esto es así ya que la ecuación de sensibilidad y ruido térmico queda definida de la siguiente forma:

$$Noise\ Floor = -174 + 10\log_{10}(1.5 * B_w) + D_{SNR} + NF$$

Siendo:

-174, son los dBm correspondientes al noise floor (límite del ruido térmico electrónico)

$(1.5 * B_w)$  es el factor de ruido añadido por Hz. Es decir, cada Hz que se consume del espectro supondrá una introducción de ruido al filtrar

$D_{SNR}$  Nivel de señal a ruido deseada para que el sistema funcione correctamente. En caso de usar una modulación GBFSK hablaríamos de 10dB aproximadamente.

$NF$  Figura de Ruido del receptor. Al menos 6 dB. Dependiente de la temperatura del sistema, algo no crítico en nuestro caso.

Si en la ecuación anterior sustituimos los valores por los conocidos en cuanto a velocidad de transmisión (1.2kbps, 50 kbps, 100kbps) podemos construir la Tabla 10:

Tasa Transmisión de Datos	Noise Floor (mínima RX)
1.2 kbps	-128 dBm
50 kbps	-108 dBm
100 kbps	-105 dBm

Tabla 10 Sensibilidad RX vs tasa de transmisión de datos

Se puede apreciar que estos datos coinciden prácticamente con los ofrecidos por el proveedor de transceptores. En función de lo bueno o malo que sea el diseño del dispositivo radio, se puede ganar algo de sensibilidad. Trabajando con una tasa de transmisión de datos de 40 kbps como es nuestro caso, la potencia mínima que puede recibirse sería aproximadamente de -110 dBm, por lo que basta con sustituir en la ecuación anterior.

Para ello se ha realizado una serie de simulaciones con el objetivo de facilitar las posibles variaciones del sistema como velocidad de transmisión, directividad de la antena etc. Para el sistema, la siguiente gráfica describe la evolución de la potencia recibida con el paso de la distancia entre Tx y Rx. El objetivo es evaluar cuál es el comportamiento en un rango entre 0 y 200 metros, considerándose ese último valor una cobertura más que suficiente.

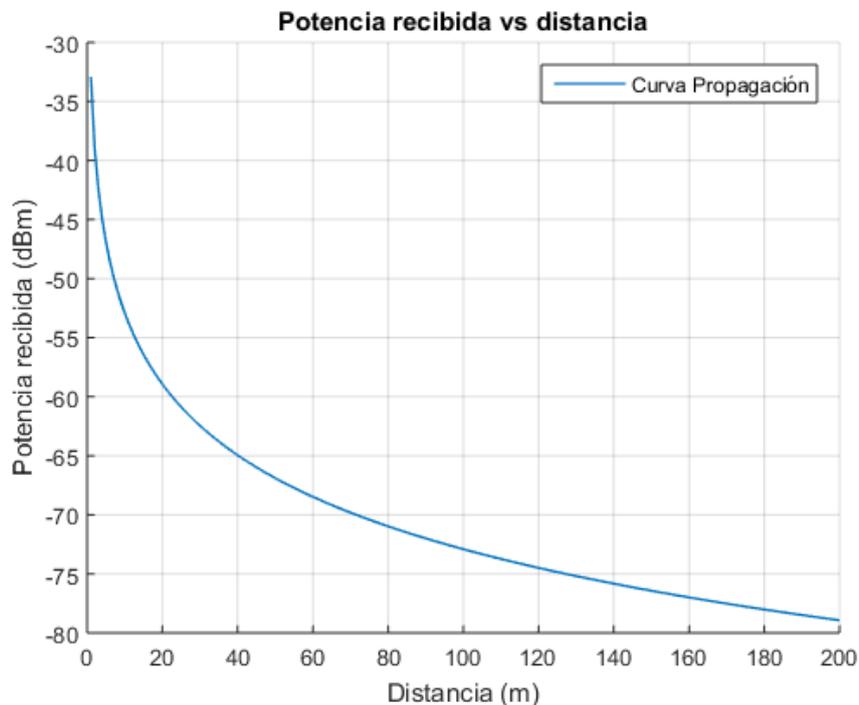


Figura 73 Curva de potencia recibida vs distancia recorrida hasta 200 m

Como se puede apreciar en la Figura 73, a 200 metros se dispondrá de una potencia de recepción teórica de -80 dBm aproximadamente. Esto deja un margen de 30 dB suficiente para garantizar una correcta comunicación a una distancia de 200 metros.

A continuación se recogen los códigos correspondientes a las simulaciones realizadas.

## Código Simulación Radioenlace

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Simulacion de RadioEnlace GFSK %%%%%%%%%%%
%Posibilidad de añadir interferencias, ruido, medidas de BER, etc.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
clc

fc = 868; %%frecuencia portadora ficticia de 900 Hz, que realmente
seria MHz.
f_data = 40;
fs = 10*fc; %%frecuencia de muestreo ficticia
length_signal = 1e4; %%enviamos 1000 simbolos
bits = round(rand(1,length_signal));
bits_NRZ = 2*bits-1; %Secuencia aleatoria de bits [-1,1]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONFORMACIÓN DEL SIMBOLOS %%%%%%%%%%%
%LAS MODULACIONES GFSK EMPLEAN UN PULSO CONFORMADOR GAUSSIANO EN VEZ
UN PULSO
%RECTANGULAR, PARA DISMINUIR LA EXTENSIÓN DEL ANCHO DE BANDA DE LA
SEÑAL A
%MODULAR EVITANDO ASÍ INTERFERENCIAS ENTRE AMBOS SÍMBOLOS.

%filtro gaussiano del GFSK para varios anchos de banda. El chip cc1200
utilzia
%por defecto un Bt = 0.5, lo que equivale a: a = 2*Ts.

filtro_conformador = ones(1,fs/f_data);
%%simbolos = filter(pulso,1,upsample(bits(1:10),fs/f_data,0));
simbolos =
filter(filtro_conformador,1,upsample(bits_NRZ,fs/f_data,0));
%simbolos_gauss = filter(pulso,1,simbolos);
%figure
%plot(simbolos)
data_modul = sin(2*pi*(fc+f_data/2*simbolos)./fs.*[0:length(simbolos)-
1]);

%%%%% AJUSTE DE POTENCIA %%%%
pot_max_dBm =14 %%Potencia máxima permitida por normativa en dBm
pot_max_W = 10^((pot_max_dBm-30)./10) %% -30 para conversión dBm-dB.
data_modul = sqrt(2*pot_max_W)*data_modul;
%data_modul = data_modul;
%figure
%plot(data_modul(1:2500))
%ylim([-1.25,1.25])
%grid on

freq_data = sqrt(2)*fft(data_modul)./length(data_modul);
figure
plot(linspace(0,fs/2,length(freq_data)/2
+1),20*log10(abs(freq_data(1:end/2 +1 ))));
title('Espectro')
xlabel('Frecuencia (MHz)')
ylabel('Potencia T_x(dB)')
grid on

```

**Código curvas BER**

```

clear all
close all
clc

SNR = [1:20];

for i = 1:length(SNR)

    fc = 900;
    f_data = 100;
    fs = 4*fc; %%frecuencia de muestreo ficticia
    length_signal = 1e5; %%enviamos 1000 simbolos
    rand('seed',-1);
    bits = round(rand(1,length_signal));
    bits_NRZ = 2*bits-1; %%Secuencia aleatoria de bits [-1,1]

    filtro_conformador = ones(1,fs/f_data);
    %%simbolos = filter(pulso,1,upsample(bits(1:10),fs/f_data,0));
    simbolos =
filter(filtro_conformador,1,upsample(bits_NRZ,fs/f_data,0));
    %%simbolos_gauss = filter(pulso,1,simbolos);
    %%figure
    %%plot(simbolos)
    data_modul =
sin(2*pi*(fc+f_data/2*simbolos)./fs.*[0:length(simbolos)-1]);

    %%% AJUSTE DE POTENCIA %%%
    pot_max_dBm =14; %%Potencia máxima permitida por normativa en dBm
    pot_max_W = 10^((pot_max_dBm-30)./10); %% -30 para conversión
dBm-dB.
    data_modul = sqrt(2*pot_max_W)*data_modul;
    %%data_modul = data_modul;
    %%figure
    %%plot(data_modul(1:2500))
    %%ylim([-1.25,1.25])
    %%grid on

    freq_data = sqrt(2)*fft(data_modul)./length(data_modul);
    %% figure
    %% plot(linspace(0,fs/2,length(freq_data)/2
+1),20*log10(abs(freq_data(1:end/2 +1))));
    %% title('Espectro')
    %% xlabel('Frecuencia (MHz)')
    %% ylabel('Potencia T_x(dB)')
    %% grid on

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Respuesta del canal %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    rx = awgn(data_modul,SNR(i)+3); %% máxima tres ya que la
distribución de ruido
    %%no se mide en la zona de frecuencias negativa. Alpha/2
    %%rx = data_modul; %%supuesto ideal sin pérdidas

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Recepción %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    filtro_rx = fir2(512,[0,f_data/fs,2*f_data/fs,1],[1,1,0,0]);
    %%plot(linspace(-
fs/2,fs/2,2^10),abs(fftshift(fft(filtro_rx,2^10))));

```

```

osci1 = sin((2*pi*(fc+f_data/2)/fs).*[0:length(rx)-1]);
%%Rama 1
ramal = rx.*osci1;
data1 = filter(filtro_rx,1,ramal);

%%Rama 1
osci2 = sin((2*pi*(fc-f_data/2)/fs).*[0:length(rx)-1]);
rama2 = rx.*osci2;
data2 = filter(filtro_rx,1,rama2);
demod = data1-data2;
%figure
%plot(demod)
data_rx = filter(filtro_conformador,1,demod);
% figure
% plot(data_rx(1500:15000));
% grid on
data_rx = data_rx((length(filtro_rx)-1)/2 +
length(filtro_conformador)+1:end);
muestreo = data_rx(1:fs/f_data:end);
bits_rx = zeros(1,length(muestreo));
bits_rx(find(muestreo>0)) = 1;
bits_erroneos(i) = sum(abs(bits(1:length(bits_rx))-bits_rx));

end

%%%%%%%%% CURVA BER GFSK %%%%%%%%%%
%%Permite obtener en función del BER estimado en Rx, cual es el
%%el nivel de señal a ruido que dispone en el radioenlace. Interesante
para
%%identificar el canal.

%%%%%%%%% Resultados prácticos del estudio del script RadioEnlace y
BER_FSK
load BER
figure
semilogy(BER, '-r');
axis([1,12,1e-5,1])
grid on
title('BER para modulacion GFSK empleada en el transceptor')
xlabel('Ratio Senal Ruido (SNR)')
ylabel('Probabilidad de Error (P_b)')

%%%%%%%%% Resultados teóricos

SNR_teo = 10.^([0:15]./10);
for i = 1:length(SNR_teo)
    Pb_teo(i) = qfunc(sqrt(SNR_teo(i)));
end

hold on
semilogy(Pb_teo)

legend('Curva Resultados Prácticos','Curva Teórica')

```

**Código estudio alcance**

```
clear all
close all
clc

Ptx = 14;
fosc = 868e6;
lamda = 3e8/fosc;
r = [1:200];
Gtx = 2.15;
Grx = Gtx;
PIRE = Ptx+Gtx;
L = 1./((4*pi).*r.^2);
Aeff = (lamda^2)/(4*pi);
Rx = PIRE+Grx+10*log10(L*Aeff);
M = 20;
Rx2 = Rx-M;
hold on
plot(r,Rx2,'LineWidth',1);
grid on
legend('Curva Propagación')
title('Potencia recibida vs distancia')
xlabel('Distancia (m)')
ylabel('Potencia recibida (dBm)')
```

## ANEXO G. Sistema STELARIA

---

El sistema STELARIA, descrito a continuación, incorpora tecnología que se encuentra a la vanguardia en ámbitos como la electrónica de potencia para iluminación, sistemas de comunicaciones y procesamiento de datos. Esto permite el encendido y apagado programado de las luminarias, la regulación dinámica de los niveles de iluminación, la representación gráfica de las luminarias sobre mapa, la monitorización en tiempo real así como el mantenimiento preventivo de cada uno de los puntos de luz instalados de forma individualizada y en múltiples instalaciones al mismo tiempo.

Del mismo modo, este sistema permite la interoperabilidad con otros sistemas de adquisición de datos y plataformas de gestión de servicios urbanos. Esto es un hecho clave en el entorno tecnológico actual del sector de la iluminación en particular y de los servicios urbanos en general.

### **Estructura del Sistema**

La estructura del sistema queda dividida en diversas capas y niveles que se presentan a continuación. En primer lugar, la capa inferior correspondiente a la luminaria y su control. Posteriormente, la red de comunicaciones y la capa de procesamiento de datos en nube, que permite una gestión de la capa inferior de forma remota. Por último, la capa superior, la aplicación web que permite trasladar la información recibida al usuario y ofrece un interfaz intuitivo para actuar sobre la red desplegada. Existe en todo momento una capa transversal a todas las descritas anteriormente, la capa de seguridad, que aporta fiabilidad y seguridad al sistema.

### **Capa Inferior. Control de la Luminaria y Dispositivos.**

La capa inferior se corresponde con la luminaria y su control, constituyendo uno de los principales factores diferenciadores de la solución. STELARIA basa sus capacidades de monitorización y control sobre los drivers para luminarias LED que ELT diseña y fabrica. El desarrollo de nuevos drivers con importantes capacidades de monitorización y control, permiten la adquisición de datos valiosos para una gestión de la iluminación urbana eficiente.

Para que todo lo mencionado anteriormente sea posible, el equipo de control de la luminaria (driver o convertidor LED) debe incorporar capacidad de adquisición, almacenamiento y tratamiento de datos. Para ello se deben emplear arquitecturas digitales basadas en microprocesador disponiendo de la capacidad de transmitir información de forma bidireccional con el exterior del equipo mediante un dispositivo y una arquitectura de comunicaciones.

Aquellos drivers o convertidores LED que no incorporen capacidad de almacenamiento y comunicación digital, no son los más apropiados para formar parte de un sistema de iluminación inteligente. Los equipos habituales que se encuentran en el mercado están dotados de una entrada de regulación analógica (0-10V, 1-10V). Este tipo de equipos no son adecuados para este tipo de sistemas dada su falta de capacidad de almacenamiento

y procesado de datos. Además no se trata de un canal de comunicaciones bidireccional, por lo que en tal caso tan solo se podría actuar sobre el driver pero no obtener ningún tipo de información referente a éste.

La tecnología de drivers eSMART desarrollada por ELT dispone de capacidad de generar perfiles de iluminación programables y en tiempo real de acuerdo con una localización de la luminaria, consiguiendo una optimización en el consumo energético y su coste asociado. Proporcionan información para el mantenimiento preventivo de instalaciones mediante parámetros del propio del driver, el módulo LED o la red eléctrica, posibilitando procedimientos de diagnóstico y test. Del mismo modo, es posible conectar a dichos drivers y gestionar sensores directamente relacionados con la iluminación como son detectores de presencia o foto-sensores



Figura 74 Driver ILCPRO 75W de ELT con tecnología eSMART

### Red de Comunicaciones

El sistema STELARIA está orientado en su arquitectura a la evolución de las luminarias como centro servidor de datos. Esto implica que los puntos de luz sean gestionables desde una infraestructura disponible en todo momento, en nube, y controlada de forma remota desde cualquier lugar.

El dispositivo de STELARIA es fácil de instalar (ver Figura 75), configurar y mantener, compatible con los diferentes modelos de luminaria que conviven en el mercado. El sistema está basado en estándares digitales entre el driver y el dispositivo de comunicaciones.

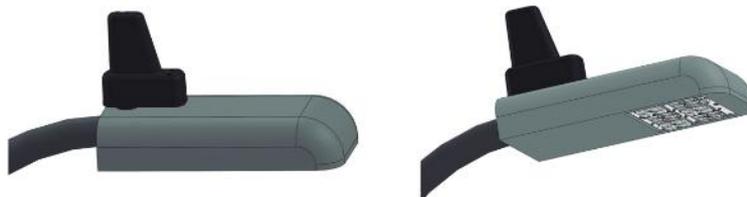


Figura 75 Integración de dispositivo STELARIA sobre luminaria

El sistema emplea tecnologías de radio robustas y probadas, funcionales en cualquier parte del mundo atendiendo a criterios de propagación radioeléctrica. Se emplea

tecnología basada en la red de telefonía móvil (2G/3G/4G) dada su extensión y uso global. La capacidad de evolución hacia modelos de conectividad flexibles basados en tecnología eSIM (Embedded-SIM) o el desarrollo de soluciones NB-IoT (Narrow Band Internet of Things) son alguno de los motivos por lo que seleccionar la banda de telefonía móvil como canal de conexión a Internet.

Se proporciona soporte técnico, servicio de ingeniería, puesta en marcha y mantenimiento de la red de comunicaciones. El objetivo fundamental de esto es tratar de no trasladar la complejidad del sistema a los gestores de la iluminación. STELARIA incorpora las tecnologías mencionadas anteriormente mediante nodos con acceso a la red de telefonía móvil mediante eSIM. Estos nodos, denominados nodos concentrados (Gateways), se encargan de la gestión de los nodos restantes que no disponen de este acceso a Internet mediante un enlace basado en radiofrecuencia.

### Capa Intermedia. Módulo de Comunicaciones y Procesado de Datos

El sistema incorpora arquitecturas nube con diferentes tecnologías de implementación compatibles e interoperables. Para ello dispone de un módulo interno de mensajería (comunicaciones), encargado de recibir los datos de cada instalación desplegada. Este módulo de comunicaciones procesa los datos para la gestión de la iluminación con mensajes de gestión de red, utilizando protocolos específicos para cada tarea como CoAP (Constrained Application Protocol) y LWM2M (Lightweight M2M). y LWM2M (Lightweight M2M).

Todo este procesado queda sustentado por un sistema de almacenamiento y persistencia de datos, basado en estructura SQL (Structured Query Language) y NoSQL que permite el procesado mediante herramientas Big Data y el intercambio de datos con otros sistemas y aplicaciones. Se dispone del mismo modo, de un módulo de gestión que presenta al usuario la información necesaria para la gestión de la iluminación, la red de comunicaciones y el envío a la capa superior de presentación. Los módulos mencionados se recogen de forma gráfica y visual en la Figura 76 que se muestra a continuación:

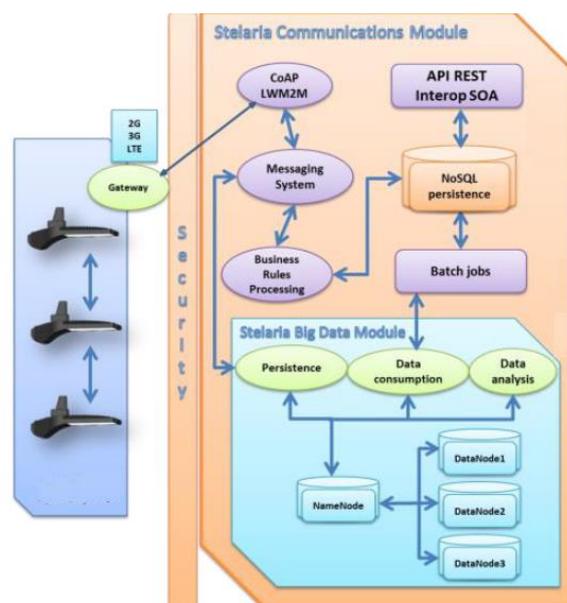


Figura 76 Módulo de comunicaciones y procesado de datos

## Capa Superior. La aplicación y Módulo de Gestión

Del mismo modo que se destaca la importancia que reside en la adquisición y procesado de datos, cabe destacar la importancia de presentar los resultados obtenidos, ofreciendo la capacidad de mostrar la información necesaria para tomar decisiones.

Se dispone de una aplicación de gestión de la infraestructura de luminarias (ver Figura 77). Se trata de una aplicación de usuario con distintos módulos funcionales, de gestión web, multidispositivo y con diferentes perfiles de usuarios. La aplicación ofrece una representación geolocalizada de los puntos de luz instalados, basada en tecnologías como Java o herramientas de última generación como Vaadin.



Figura 77 Software de monitorización y gestión. Interfaz Web multi-dispositivo

Así mismo, se dispone de un módulo de interoperabilidad con otras aplicaciones, sistemas o plataformas smart city o de gestión de ciudad, basadas en estructuras SOA (Service Oriented Architecture), RESTful (Representational State Transfer) y concepto de servicio web. La Figura 78 muestra la arquitectura software, donde destacan los módulos de gestión e interoperabilidad mencionados.

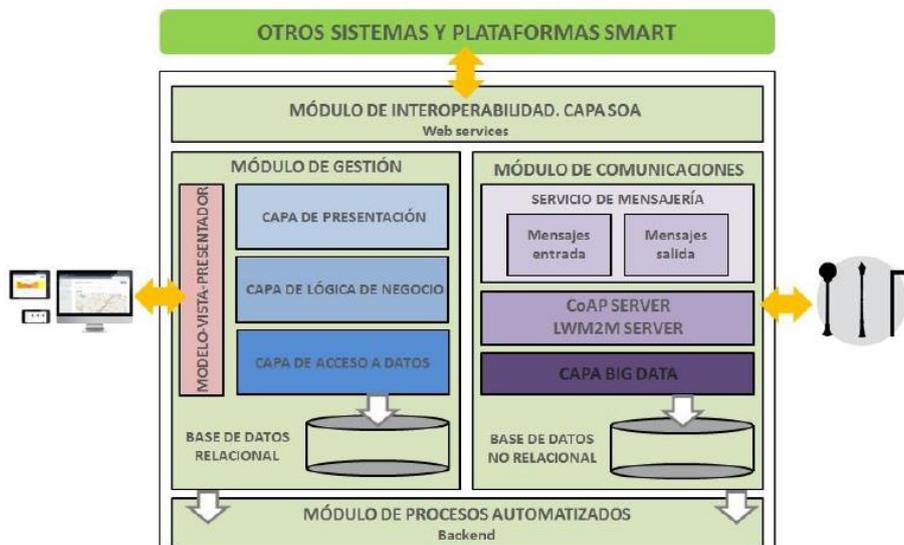


Figura 78 Arquitectura software. Módulo de gestión, módulo de interoperabilidad y aplicación de usuario (modelo vista-presentador)

## Capa Transversal. La Seguridad

El objetivo de esta capa es garantizar la seguridad, privacidad y fiabilidad del sistema. Para ello, se han implementado diferentes niveles y mecanismos de protección para que las comunicaciones y los datos sean seguros de extremo a extremo, tales como encriptación AES (Advanced Encryption Standard) 128 bits en las capas más bajas de la comunicación o SSL y HTTPS en el intercambio de datos con la nube. Además se han añadido sistemas en espejo a diferentes niveles o control y gestión de permisos de acceso al sistema a nivel de aplicación.

Con el fin de garantizar la autenticación y autorización de usuario de la aplicación, STELARIA soporta múltiples modelos de autenticación como HTTP básico, HTTP Digest, LDAP, OPENID, JAAS etc.

## Funcionalidades del Sistema

En este apartado se recogen las funcionalidades más relevantes del sistema de gestión inteligente de alumbrado STELARIA. La aplicación de usuario queda dividida en tres bloques principales. En primer lugar, el módulo de administración de la aplicación. Seguidamente el módulo de gestión de iluminación y finalmente el módulo de gestión de la red de nodos.

Este primer bloque, correspondiente al módulo de administración de la aplicación, tiene como misión hacer posible la configuración de todos aquellos elementos que ofrece la aplicación referente a la creación y modificación de usuarios, roles, ubicaciones, permisos, instalaciones, calendarios etc.

El segundo bloque, referente a la gestión de la iluminación, permite en base a los permisos de usuario disponibles, realizar las tareas relacionadas con la gestión de las luminarias tales como su localización geográfica o agrupaciones lógicas definidas. Del mismo modo permite crear modificar y aplicar programas de alumbrado, escogiendo la opción más idónea en base a criterios como el modelo, la potencia consumida, el barrio, las coordenadas geográficas etc. Así mismo, el sistema permite realizar una monitorización y almacenamiento de datos durante todo el periodo de vida operativo de la instalación, de parámetros relevantes como:

- Identificación, estado y posición en mapa de cada luminaria perteneciente a la instalación
- Definición y reporte de eventos y alarmas de cada luminaria perteneciente a la instalación.
- Monitorización de los parámetros más relevantes de la luminaria tales como tensión, intensidad, temperatura y potencia.
- Monitorización de los consumos exactos de cada luminaria.
- Monitorización del estado de la vida operativa de la luminaria y sus componentes.
- Definición de agendas de mantenimiento preventivo y reactivo.

Por último, el módulo de gestión de la red de nodos. A través del mismo se podrá monitorizar y actuar sobre la red de comunicaciones durante el proceso de instalación de la red de luminarias así como durante el transcurso de la vida útil.

Las principales funciones de este bloque son las siguientes:

- Identificación, información acerca del estado y localización de cada nodo presente en la red (ver Figura 79)
- Información de consulta relativa a la gestión electrónica del nodo (temperatura, tensión, estado de la conexión del equipo de potencia etc)
- Información de consulta de los parámetros más relevantes referentes a la red de comunicaciones (RSSI (Received Signal Strength Indicator), LQI (Link Quality Indicator), ETX (Expected Transmisión Count) entre otros).
- Gestión del firmware de los nodos y actualizaciones (FOTA (Firmware Over the Air))

## Mapa de nodos

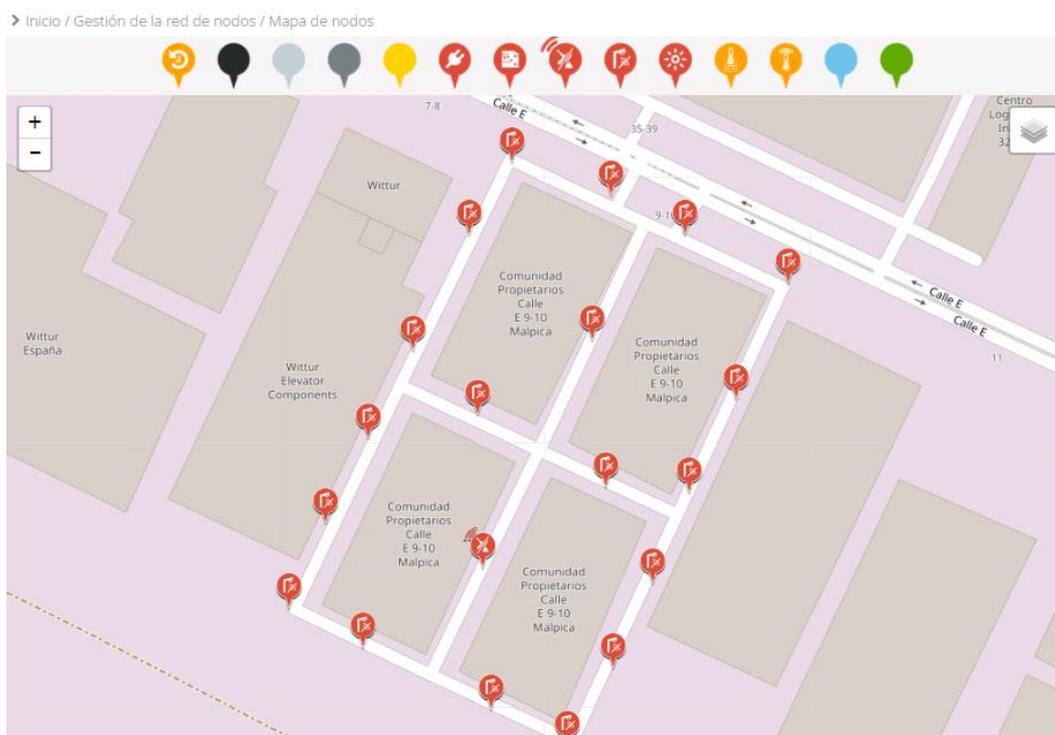


Figura 79 STELARIA. Ejemplo de localización y estado de cada luminaria instalada

## Interoperabilidad del Sistema

Una vez que se dispone de la capacidad de transportar y tratar datos del alumbrado público, es posible combinarlos con otras fuentes de datos urbanos creando herramientas de análisis y soluciones de valor añadido para las ciudades y gestores municipales.

STELARIA incorpora dicha plataforma de almacenamiento y procesamiento de datos, con la correspondiente capacidad de inteligencia de negocio y herramientas de presentación y visualización de resultados. Esto permite la combinación de datos correspondientes al

alumbrado público con datos pasivos de los sistemas de gestión municipal como residuos, agua, tráfico, emergencias, e-administración etc.

ELT ha llevado a cabo el proyecto SLUX (Smart Lighting For Urban Experience), que tiene por objetivo proporcionar un alumbrado eficiente, inteligente, dinámico y adaptable en tiempo real en base a las necesidades de los ciudadanos. Para ello, ha desarrollado e incorporado a STELARIA un módulo específico que combina los datos activos obtenidos de instalación de alumbrado público, con datos pasivos de los sistemas de gestión municipal, tales como padrón, contabilidad, inventario etc.

Esto permite adaptar el alumbrado de un área concreta en base a la cantidad de personas que habitan en una ciudad concreta, su edad, sexo etc. En la Figura 80 se muestra un ejemplo de la información obtenida como nexo de ambas plataformas (alumbrado y padrón) para el caso de una instalación real.

## Indicadores de ámbito demográfico

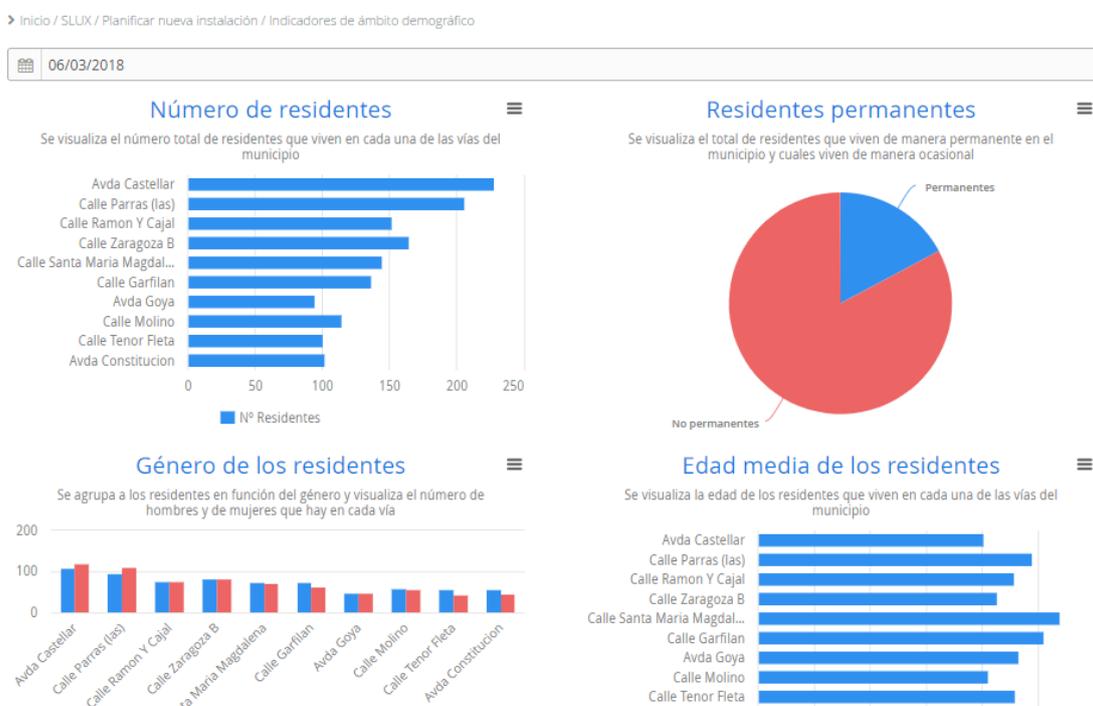


Figura 80 Estadísticas de un ejemplo de sistema de gestión municipal de padrón

Otro indicador fundamental de cara a decidir cuál debe ser la iluminación de una calle es la actividad predominante de la zona. Esto influye de forma determinante en los horarios escogidos para el encendido y apagado así como para las reducciones de flujo y su intensidad. A modo de ejemplo, se puede comparar la diferencia que existe en las necesidades de alumbrado para una zona residencial y una zona de negocios o industrial. La Figura 81 muestra la herramienta visual desarrollada por ELT que permite analizar el tipo de actividad para una localidad concreta.

Desde un punto de vista de análisis económico, se puede usar este ejemplo de interoperabilidad para evaluar el impacto de la inversión del alumbrado en los ciudadanos

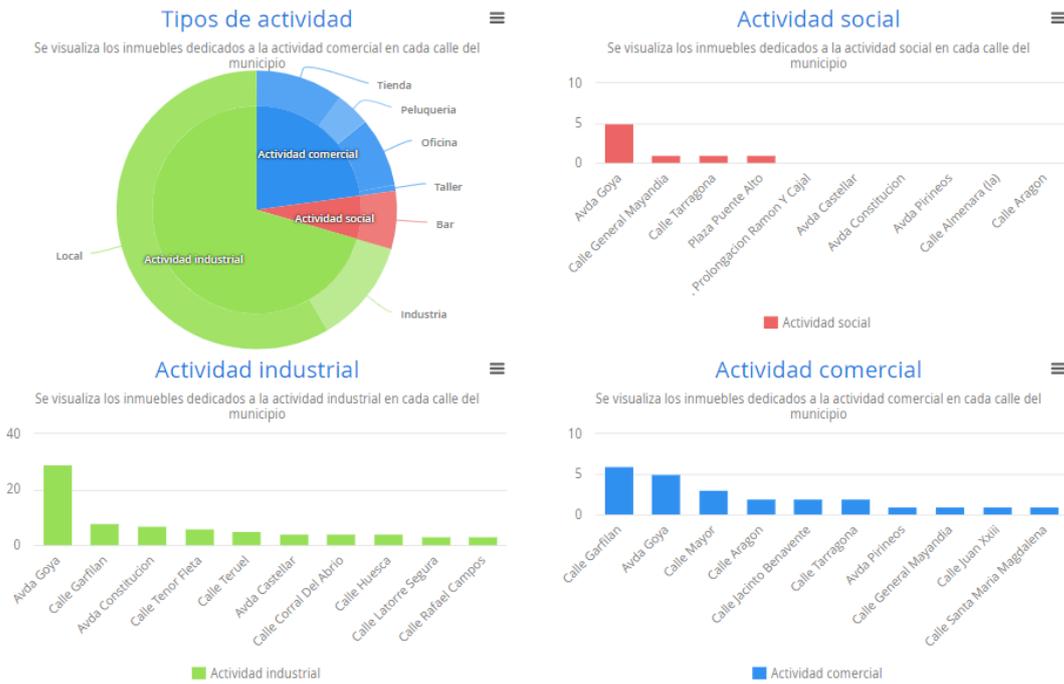


Figura 81 Herramienta de visualización de tipo de actividad para una localidad

Todo ello facilitará al responsable de la gestión de alumbrado la elección del alumbrado más adecuado. No solo será una elección desde un punto de vista de aspecto físico (tipo de luminaria, flujo lumínico, rendimiento etc.), sino desde un punto de vista funcional (valores e intervalos de regulación, horario de funcionamiento). La Figura 82 muestra la herramienta de planificación de instalación, que utiliza toda la información previamente descrita y permite obtener una decisión referente a la instalación en base a una ponderación de todos los parámetros mencionados.

Sistema de planificación de instalaciones

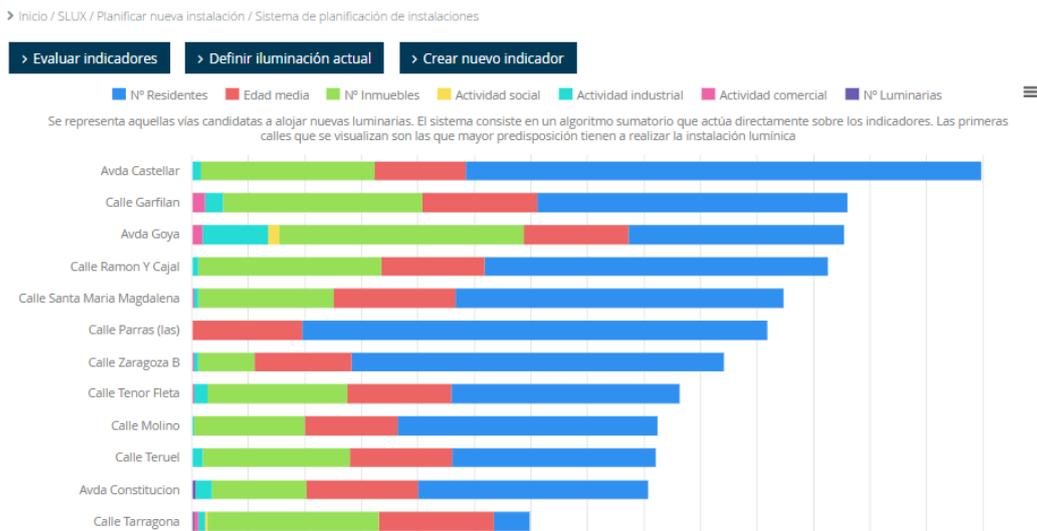


Figura 82 Sistema de planificación de instalaciones. SLUX

## Ejemplo del Sistema STELARIA

A continuación se recogen las imágenes correspondientes a la información recogida por la plataforma STELARIA. Se muestra el histórico de datos de forma gráfica para una luminaria situada en una instalación real donde se parametriza continuamente parámetros de gran relevancia como la tensión de red eléctrica, el consumo energético del punto de luz o la temperatura del driver.

Como se puede apreciar en la Figura 83 referente al consumo energético generado, el driver está ejecutando un perfil de iluminación que durante las horas menos necesarias de iluminación realiza una reducción del flujo lumínico, conllevando esto un recorte considerable del consumo energético y con ello del gasto económico.

Del mismo modo se puede apreciar en la temperatura del driver como existe un regimen transitorio desde los 20°C hasta los 45°C. Esto nos indica que la luminaria ha realizado un encendido instantes previos al rango de hora mostrado.

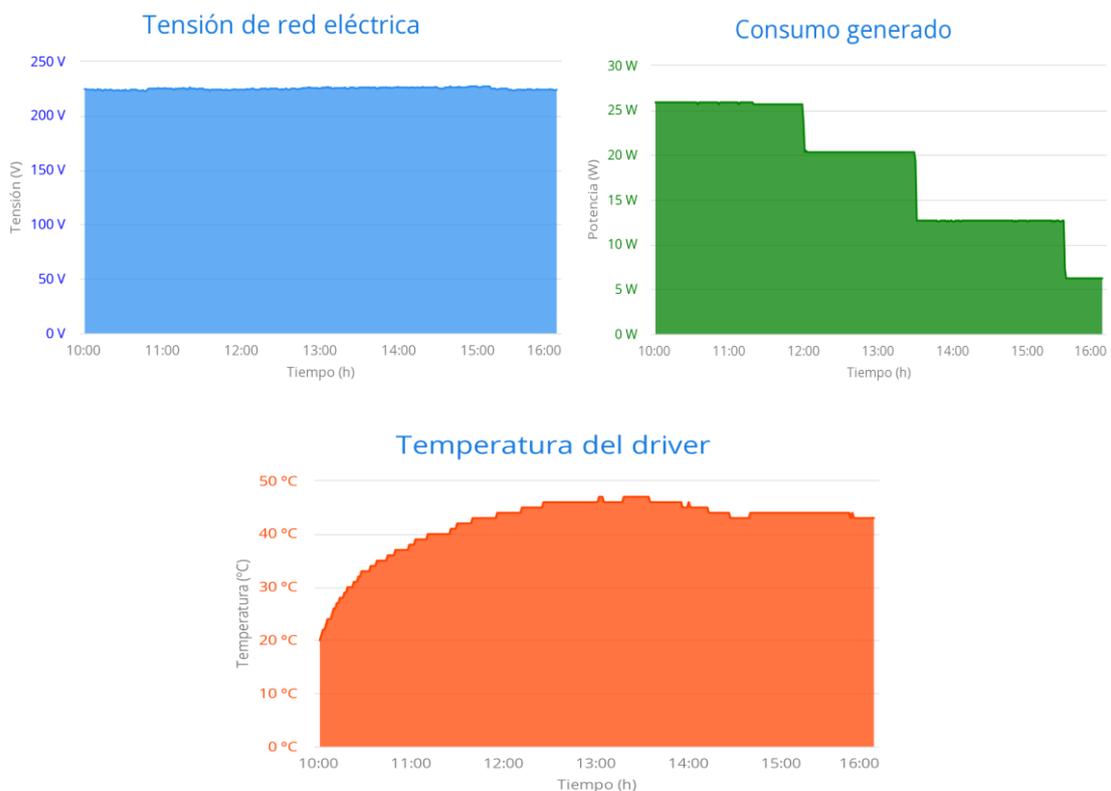


Figura 83 Gráficas referentes a un punto de luz instalado

ANEXO H. Hojas de Características

2.1

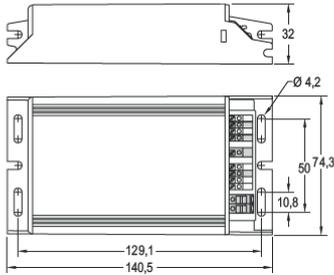
DRIVERS  
CONSTANTE  
CORRIENTE

**iLC PRO-XR**

180-277V  
AC 50...60Hz



Full PROGRAMMABLE control gear up to 75W. IP20  
Equipo PROGRAMABLE hasta 75W. IP20



Full Programmable electrical parameters and functionalities: AOC (Adjustable Output Current), MTP (Module Temperature Protection), CLO (Constant Lumen Output), EOL (End Of Life module alarm), PST (Programmable Start-up Time).  
Interfaces: DALI, 0-10V, 1-10V, ActiDIM, ActiDIM + Parking, MainsDIM, LineSwitch and Parking.  
Compatible version with STELARIA® Remote Wireless Street Lighting CMS available.  
Double or reinforced insulation control gear, for built-in-use. Protection index IP20.  
Output constant current allowed: 70 ... 1400mA.  
Maximum output power: 75W.  
Output ripple current (ORC) <5%.  
Permitted input voltage AC: 162 ... 305V.  
Low Total Harmonic Distortion (THD @230Vac, 75W) <8%.  
Dimming range: 100% down to 5% (minimum output current = 70mA).  
Protection against short circuit, overload and no load operation.  
Control gear thermal protection.  
External LED module thermal protection connector.  
Mains surge protection integrated:  
- Differential mode: 6kV / 3kA (L-N).  
- Common mode: 8kV (L/N-Earth).  
Lifetime up to 100.000h\*.  
Electronic circuit fully protected against humidity.  
High quality light without flickering.  
Low Stand-by power consumption: < 0,5W.

Parámetros eléctricos y funcionalidades programables: AOC (corriente de salida ajustable), MTP (protección térmica del módulo), CLO (compensación del flujo luminoso), EOL (alarma de fin de vida del módulo), PST (tiempo de encendido programable).  
Interfaces: DALI, 0-10V, 1-10V, ActiDIM, ActiDIM + Parking, MainsDIM, LineSwitch y Parking.  
Versión disponible compatible con sistema de gestión remota de alumbrado STELARIA®. Equipo con aislamiento doble e reforzado, a incorporar. Índice de protección IP20.  
Rango de corriente de salida: 70 ... 1400mA.  
Máxima potencia en la salida: 75W.  
Rizado de corriente de salida (ORC) <5%.  
Tensión de entrada permitida AC: 162 ... 305V.  
Baja distorsión armónica (THD @230Vac, 75W) <8%.  
Rango de regulación: 100% hasta 5% (corriente de salida mínima = 70mA).  
Protección contra cortocircuito, sobrecarga y en circuito abierto.  
Protección térmica en el equipo electrónico.  
Conexión para protección térmica del módulo LED.  
Protección contra sobretensiones de red integrada:  
- Modo diferencial: 6kV / 3kA (L-N).  
- Modo común: 8kV (L/N-Tierra).  
Vida útil hasta 100.000h\*.  
Circuito electrónico protegido contra la humedad.  
Elevada calidad de la luz sin parpadeos.  
Bajo consumo en Stand-by: < 0,5W.

User guide on  
[http://www.elt.es/productos/esmart\\_en.html](http://www.elt.es/productos/esmart_en.html)  
Packaging and weight pag. 238 and  
[www.elt.es/productos/packaging\\_ELT.pdf](http://www.elt.es/productos/packaging_ELT.pdf)  
Instructions manual on  
[www.elt.es/productos/inst\\_manual.html](http://www.elt.es/productos/inst_manual.html)

Guía de usuario en  
[http://www.elt.es/productos/esmart\\_es.html](http://www.elt.es/productos/esmart_es.html)  
Embalaje y peso pag. 238 y  
[www.elt.es/productos/packaging\\_ELT.pdf](http://www.elt.es/productos/packaging_ELT.pdf)  
Manual de instrucciones en  
[www.elt.es/productos/manual\\_instrucciones.html](http://www.elt.es/productos/manual_instrucciones.html)

Technical characteristics / Características técnicas

Ref. No.	Output power range / Rango de potencia en módulo	Output current / Corriente de salida	Output voltage range / Rango de tensión de salida	Power factor at full load / Factor de potencia a carga máxima	Max. system efficiency / Rendimiento máx. del sistema	Max. temp. at te point / Temp. máx. envolvente	Min. Operating temp / Temp. mín. funcionamiento
	W	mA	Vdc	$\lambda$	$\eta$ (%)	tc (°C)	ta (°C)
iLC PRO 75/200...1400-XR 9916151	7,4... 37,8	350	21... 108	0,98	>89	90	-40
	10,5... 54	500	21... 108		>90		
	14,7... 75	700	21... 108		>90		
	22... 75	1050	21... 72		>91		
	25,2... 70	1200	21... 58		>90		
	29,4... 70	1400	21... 50	>88			

Logistical data / Datos logísticos - P. 238

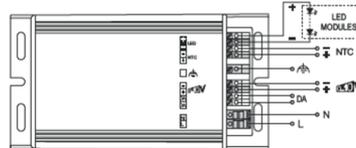
	Net unit weight / Peso neto unitario	Units per box / Unidades por caja
iLC PRO-XR	0,436 Kg	8 uds.

Accessories / Accesorios



iProgrammer

Pag. 35



0617-02



32



OUTDOOR SERIES / 2. TECHNICAL LIGHTING - CONSTANT CURRENT / 2.1 DRIVERS  
SERIE OUTDOOR / 2. ALUMBRADO TÉCNICO - CORRIENTE CONSTANTE / 2.1 DRIVERS

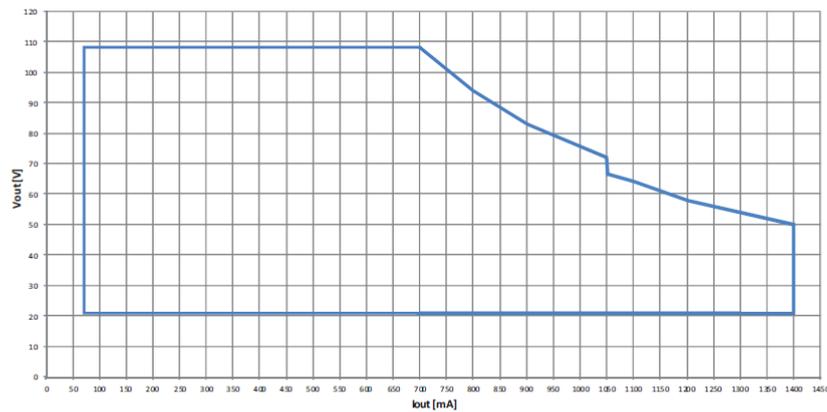


# iLC PRO-XR

180-277V  
AC 50...60Hz

Full PROGRAMMABLE control gear up to 75W. IP20  
Equipo PROGRAMABLE hasta 75W. IP20

### Operating area / Área de operación



Adjustable Output Current (AOC) / Corriente de salida ajustable (AOC)	Regulation / Regulación	Minimum output voltage / Tensión de salida mínima	Maximum output voltage / Tensión de salida máxima	Minimum module power / Potencia mínima del módulo	Maximum module power / Potencia máxima del módulo
mA		V	V	W	W
70... 199	ON/OFF	21	108	$\frac{AOC (mA) \times 21}{1000}$	$\frac{AOC (mA) \times 108}{1000}$
200... 700	✓	21	108	$\frac{AOC (mA) \times 21}{1000}$	$\frac{AOC (mA) \times 108}{1000}$
701... 1050	✓	21	$\frac{75 \times 1000}{AOC (mA)}$	$\frac{AOC (mA) \times 21}{1000}$	75
1051... 1400	✓	21	$\frac{70 \times 1000}{AOC (mA)}$	$\frac{AOC (mA) \times 21}{1000}$	70

### Factory default configuration / Configuración de fábrica

Enabled functionalities / Funcionalidades habilitadas

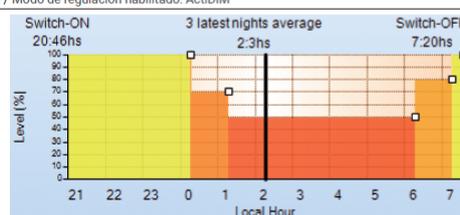
Adjustable Output Current (AOC) Corriente de salida ajustable (AOC)	700 mA
Module Temperature Protection (MTP) Protección térmica del módulo LED (MTP)	✗
Constant Lumen Output (CLO) Compensación del flujo luminoso (CLO)	✗
End-Of-Life module alarm (EOL) Alarma fin de vida del módulo (EOL)	✗
Programmable Start-up Time (PST) Encendido programable (PST)	✗



### Factory default configuration / Configuración de fábrica

Enabled regulation mode: ActiDIM / Modo de regulación habilitado: ActiDIM

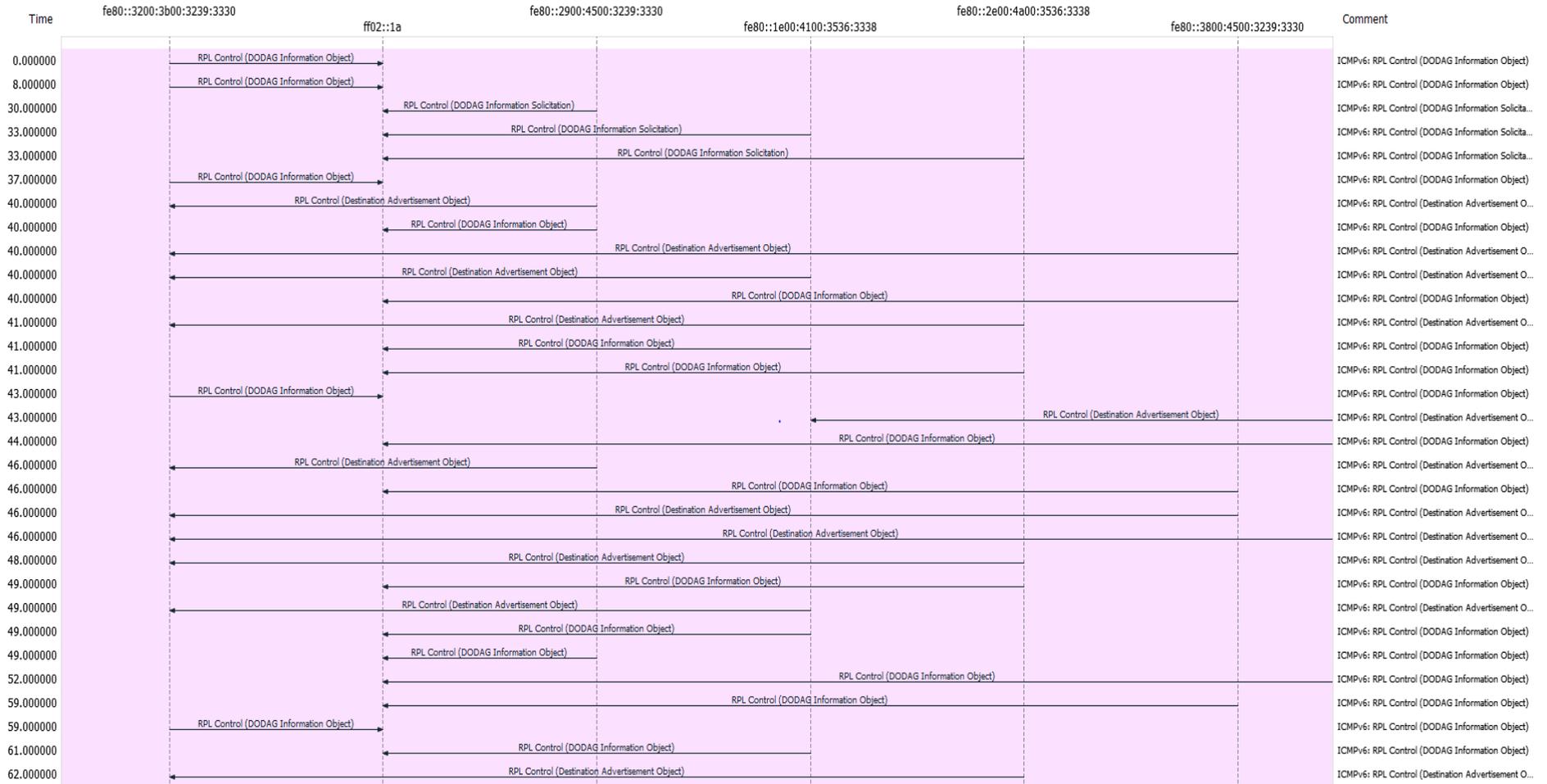
Time intervals / Intervalos temporales	Module power / Potencia del módulo
Power on / Encendido	100%
2 hours before the middle of the night / 2 horas antes de mitad de la noche	70%
1 hour before the middle of the night / 1 hora antes de mitad de la noche	50%
4 hours after the middle of the night / 4 horas después de mitad de la noche	80%
5 hours after the middle of the night / 5 horas después de mitad de la noche	100%

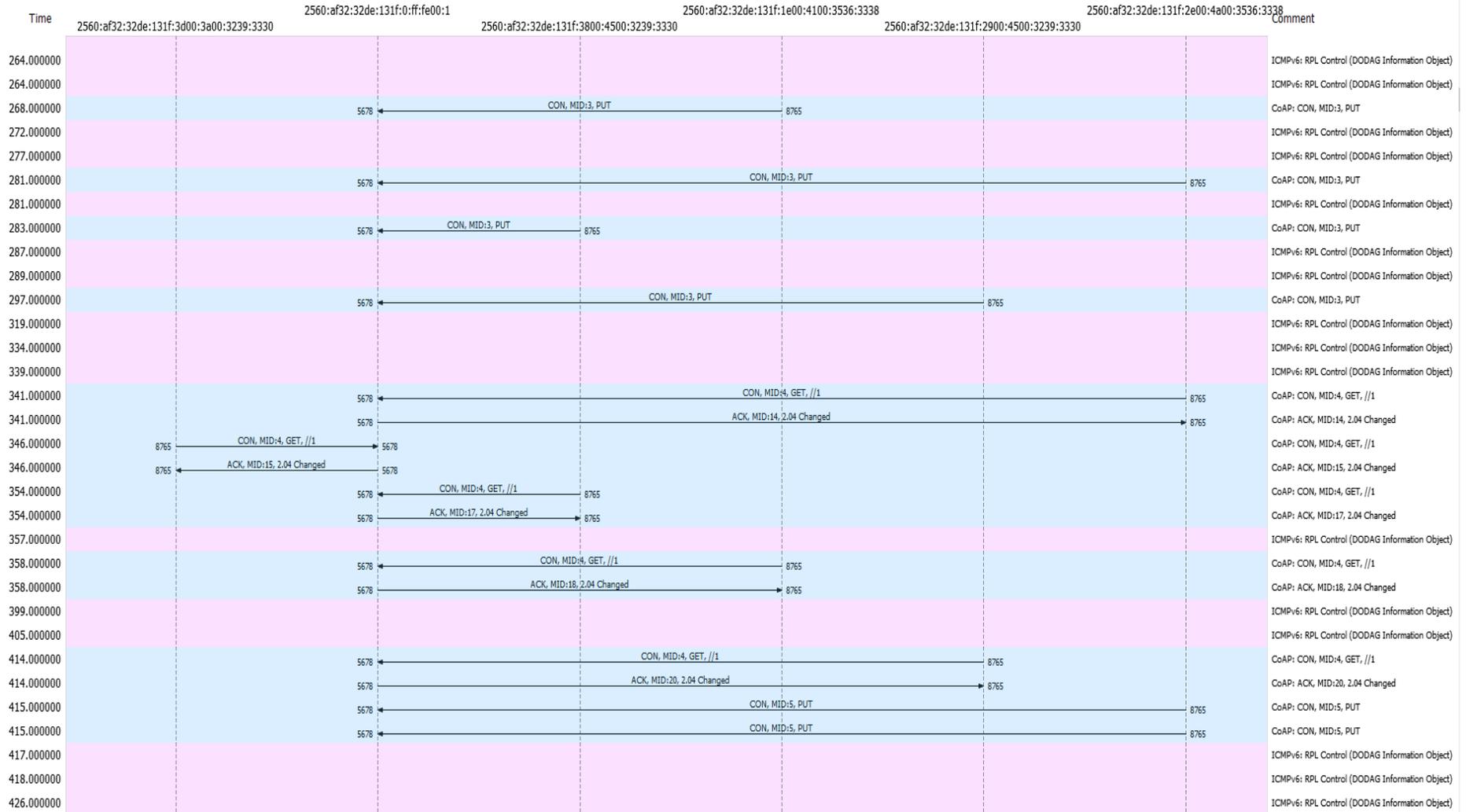


0617-02



# ANEXO I. Capturas de Tráfico en Wireshark







## Referencias

---

- [1] A. Rubio Dobón, «El alumbrado público inteligente como infraestructura base para la implantación del ecosistema IoT en el entorno rural,» 22 12 2017. [En línea]. Available: <https://www.esmartcity.es/comunicaciones/alumbrado-publico-inteligente-infraestructura-base-la-implantacion-del-ecosistema-iot-entorno-rural>.
- [2] I. Cid Jiménez, «Últimos pasos de la ledificación,» 4 1 2018. [En línea]. Available: <https://www.elt.es/blog/ultimos-pasos-de-la-ledificacion>.
- [3] R. EFEverde, «España es el país europeo que más gasta en alumbrado público,» EFEverde, 31 8 2017. [En línea]. Available: <https://www.efeverde.com/noticias/espana-pais-mas-gasta-alumbrado-publico/>.
- [4] J. J. González Méndez, «Tecnologías, arquitecturas y soluciones para la iluminación inteligente,» *XIII Congreso Panamericano de Iluminación LUXAMERICA*, 2016.
- [5] K. Pothunganti y A. Chitneni, «A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee and Wi-Fi,» *Advance in Electronic and Electric Engineering*, vol. 4, n° 6, pp. 655-662, 2017.
- [6] U. Raza, P. Kulkarni y M. Sooriyabandara, «Low Power Wide Area Networks: An Overview,» *IEEE Communications Surveys & Tutorials*, vol. 19, n° 2, pp. 855-873, 2017.
- [7] J. Burns, S. Kirtay y P. Marks, «Future use of licence exempt radio spectrum,» 14 7 2015. [En línea]. Available: [http://www.plumconsulting.co.uk/pdfs/Plum\\_July\\_2015\\_Future\\_use\\_of\\_Licence\\_Exempt\\_Radio\\_Spectrum.pdf](http://www.plumconsulting.co.uk/pdfs/Plum_July_2015_Future_use_of_Licence_Exempt_Radio_Spectrum.pdf).
- [8] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, J. Melia-Segui y T. Watteyne, «Understanding the limits of LoRaWAN,» *IEEE Communications Magazine*, vol. 55, n° 9, pp. 34-40, 2017.
- [9] T. H. Lee, H. S. Chiang, L. H. Chang y M. C. Hsieh, «Performance Analysis of IEEE 802.15.4 868MHz, 915MHz and 2.4GHz Physical Schemes in 6LoWPAN,» National Taichung University of Education, Department of Computer Science, Taichung, 2015.
- [10] Internet Engineering Task Force, «Request for Comments 6282,» 9 2011. [En línea]. Available: <https://tools.ietf.org/html/rfc6282>.
- [11] O. Jonas, «6LoWPAN demystified,» 1 10 2014. [En línea]. Available: <http://www.ti.com/lit/wp/swry013/swry013.pdf>.

- [12] Group Network Working, «RIP Version 2, Request for Comments: 2453,» 11 1998. [En línea]. Available: <https://tools.ietf.org/html/rfc2453>.
- [13] Group Network Working, «OSPF for IPv6, Request for Comments: 5340,» 7 2008. [En línea]. Available: <https://tools.ietf.org/html/rfc5340>.
- [14] Internet Engineering Task Force, «RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, Request for Comments: 6550,» 3 2012. [En línea]. Available: <https://tools.ietf.org/html/rfc6550>.
- [15] Internet Engineering Task Force, «The Constrained Application Protocol (CoAP), Request for Comments: 7252,» 6 2014. [En línea]. Available: <https://tools.ietf.org/html/rfc7252>.
- [16] A. Dunkels, O. Schmidt, T. Voigt y M. Ali, «Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded Systems,» Swedish Institute of Compute Science.
- [17] R. Khadse, N. Gawai y B. M.Faruk, «Overview and Comparative Study of Different Microcontrollers,» *International Journal for Research in Applied Science & Engineering Technology*, vol. 2, nº 12, pp. 311-217, 2014.
- [18] I. Romdhani, A. Yassin-Al-Dubai, M. Qasem y B. Ghaleb, «Cooja Simulator Manual,» Edinburgh Napier University, Edimburgo, 2016.
- [19] D. Ana, «6LoWPAN, IPv6 for Wireless Sensor Networks,» GridTICs, Mendoza, 2012.
- [20] Especialidades Luminotécnicas, «STELARIA-ELT,» [En línea]. Available: <https://www.elt.es/stelaria-sistema-inalambrico-de-gestion-remota-de-alumbrado>.
- [21] R. L. Alfonso, «Eficiencia Energéticas vs Alumbrado Público,» Phillips-Indal, Barcelona, 2012.