Universidad de Cantabria

Departamento de Ingeniería de Comunicaciones

TESIS DOCTORAL

Contribución al diseño, definición e implementación de una plataforma de investigación para la Internet del Futuro, basada en un despliegue masivo de redes de sensores inalámbricos heterogéneos, en el marco de la Ciudad Inteligente

Autor: *José Antonio Galache López*
Director: *Luis Muñoz*

Santander, octubre de 2013

Certificado del director de la Tesis

D. Luis Muñoz, Catedrático de la Universidad de Cantabria en el Área de Ingeniería Telemática

HACE CONSTAR:

Que la Tesis titulada **"Contribución al diseño, definición e implementación de una plataforma de investigación para la Internet del Futuro, basada en un despliegue masivo de redes de sensores inalámbricos heterogéneos, en el marco de la Ciudad Inteligente"**, ha sido realizada por **D. José Antonio Galache López** en el Departamento de Ingeniería de Comunicaciones de la Universidad de Cantabria bajo mi dirección, y que reúne las condiciones exigidas a los trabajos de doctorado.

Santander, octubre de 2013

Fdo. Luis Muñoz

A mi hermana, por nunca dejarme sentirme sólo.

A mi padre, por mostrarme el camino recto.

A mi madre, por no permitirme desfallecer.

AGRADECIMIENTOS

Me encuentro aquí sentado, ante el capítulo más importante de este trabajo, el más complejo porque no quisiera olvidarme de nadie (si lo hago pido disculpas), pero al mismo tiempo el más gratificante porque me presenta el marco perfecto para agradecer el apoyo a la gente que me rodea.

La primera persona a la que quiero mencionar es al director de esta Tesis, Luis Muñoz. Técnica y profesionalmente, he aprendido muchísimo de ti durante estos años, pero sin lugar a duda, me quedo con el crecimiento personal a través de la capacidad de trabajo y los valores que me has transmitido, de una forma muy sencilla pero desgraciadamente poco extendida, predicando con el ejemplo.

Mi segunda casa es el Grupo de Ingeniería Telemática, donde fui acogido hace ya más de 10 años cuando me dejé caer, un verano de 2001, para empezar a impregnarme un poco del mundo de la investigación. Por el grupo ha pasado mucha gente estos años, Ángel, Óscar, Dulce, Raquel, Laura, Paco, Chema, Miguel I., Jabo, Edgar, Noé, Juanisho, Miguel C. y muchos más, de los que siempre he intentado aprender y quedarme con lo mejor. Respecto a aquéllos con los que comparto el día a día actualmente, en primer lugar están mis compañeros del Laboratorio 14; Chus y nuestras innumerables discusiones de geografía de Cantabria (espero seguir teniéndolas) y Vero, que desde el principio siempre se ha preocupado por mí y con la que he compartido trabajo en muchos proyectos, siempre aderezados con bromas quedando momentos inolvidables. Por la puerta este del 14, se encuentran los despachos de Marta, Roberto, Jota, Klaus y Alberto (a un nivel superior), con los que he compartido muchas charlas en la travesía desde/hacia el coche o en muchos de los cafés que hemos tomado juntos. Por la salida principal del 14, el primer despacho que se encuentra es el de Laura que alguna vez habrá sufrido nuestras discusiones técnicas y no tan técnicas, luego Luis S. del que he aprendido muchas cosas trabajando a su lado y George, alguien que siempre está dispuesto a ayudarte sea cual sea el problema, dedicando el tiempo necesario (aunque su móvil se oiga sonar de fondo una y otra vez). En el siguiente despacho, Johnny con todas las expresiones que nos enseña, 'una raya más al tigre' o 'cuando el gato no está, los ratones hacen fiesta' entre las más celebres; y Món, el director de mi proyecto Fin de Carrera, con el que he compartido infinidad de dudas técnicas, resultados deportivos (en su día vía sms patrocinado y ahora con whatsapp), y siempre dispuesto a echarme un cable cuando lo necesito. Atravesando el 'jardin', llegamos al Laboratorio 21 donde, al entrar, debes evitar el triángulo tecnológico que forman Pablo, David y Juanra o, de lo contrario, puedes quedar absorbido. Pablo con sus geniales discusiones de código y de arquitectura, David con sus gráficas con todo tipo de parámetros (incluido el burglary rate), y Juanra (el androide) con el que he compartido tantos días de instalación y al que nunca he visto una mala cara ni un mal gesto, aunque la carga de trabajo fuese enorme. Si consigues pasar el triángulo, te encuentras con Pablo M., Carmen, Luisco, Nacho y Javier C., con los que tantas veces he arreglado el mundo en la cafetería de Físicas. Me queda una persona, pero aparecerá más tarde.

Otro de los pilares que me ha ayudado en esta andadura son, por supuesto, mis amigos. Entre ellos mis compañeros de carrera, como Jose al que cuesta bastante verle por la tierruca, Fernando y David con sus inseparables palas de pádel, Juan y Raquel con los que he compartido alguna que otra barbacoa, y Adolfo que siempre está dispuesto a hablar un rato y a proponer planes cuando lo necesitas. Luego están los amigos con los que quedo más a menudo, Rebe la más antigua y espero

que por mucho tiempo más, Luis y sus disparatados comentarios, Álvaro como MasterChef del grupo, Rosa siempre con una sonrisa, David y Javi siempre a su servicio (informático), Silvia buscando planes y escapadas y Marta con sus comentarios que nunca te dejan indiferente; además de Agus también conocido como el sheriff, y las hermanas González ¿quién me cae mejor hoy?. No quiero olvidarme de Quique y Mónica, Jorge y Patri, Fernando y muchos más, con los que he vivido muy buenos momentos. Respecto a los que están fuera, Héctor la verdad es que este primer verano 'sin' palas ha sido muy duro, Leyre hace mucho que no discutimos, Luis y Su os debo una visita a la capi, y otra a Fráncfort a Marta y Javi.

Por otro lado, aquellos amigos que conocí gracias a mi hermana y que ahora también se han convertido en los míos, Fredo espero que vengas pronto por aquí y nos cuentes cómo va todo, Vero ahora me pondré con el molde que lo tengo abandonado, Ingrid te pediré algún consejo facultativo y Álex a ver cuándo nos vemos que últimamente no coincidimos.

En todo este tiempo que ha pasado desde que empecé esta andadura, me ha dado tiempo a emanciparme y, circunstancias de la vida, conocer a gente extraordinaria en la Comunidad, Isaac, Laura, José Ramón y Teresa, que grandes momentos tomando una caña y charlando, habitualmente de la fachada.

A mi cuñado, Chus, con el que he compartido muchos momentos, partidos, conciertos, barbacoas, y que espero seguir compartiéndolos y disfrutándolos con él.

Finalmente, quiero expresar mi agradecimiento a las tres personas más importantes de mi vida y sin las cuáles la realización de esta Tesis jamás hubiese sido posible. A mi padre por inculcarme sus valores de rectitud aunque sean difíciles de aplicar en esta vida, a mi madre por transmitirme su perseverancia para afrontar los retos por muy complejos que éstos sean; a los dos porque me disteis la vida a mí y a Laura, que siempre ha estado a mi lado cuando la he necesitado y nunca me ha dejado sentirme sólo. Os quiero.

RESUMEN

Desde su nacimiento, la evolución de las redes de sensores, tanto en su capacidad de cómputo y almacenamiento, como en la de medida de nuevos parámetros, se ha traducido en un aumento de la complejidad y heterogeneidad asociadas a las mismas. Entre los años 2008 y 2009, cuando el número de dispositivos (cosas u objetos) conectados a través de Internet ya excedía el número de personas conectadas, se acuñó el término de la Internet de las cosas. Con el actual ritmo de crecimiento se estima que para el año 2020 varias decenas de miles de millones de dispositivos, equipados con diferentes tipos de sensores y actuadores, estarán conectados a Internet mediante redes de acceso heterogéneas. En este sentido, aspectos tales como la escalabilidad, heterogeneidad y limitaciones de los dispositivos, así como la distinta naturaleza de las interacciones entre ellos, se plantean como retos para una integración exitosa dentro de la arquitectura de la Internet del futuro.

Con estas condiciones de contorno, resulta esencial disponer de infraestructuras de experimentación, en el ámbito de la Internet de las cosas, que posibiliten investigar e innovar con las tecnologías propias de aquéllas. Este tipo de infraestructuras presenta enormes ventajas frente a entornos de laboratorio, o incluso frente a las mismas herramientas de simulación, ya que permiten trabajar con parámetros y consideraciones basadas en el estado del arte de la tecnología, a la vez que se superponen las condiciones reales de operación y las demandas de calidad de servicio requeridas. Además, cuando ese tipo de facilidades experimentales se aproximan a los usuarios finales, cabe esperar un impacto mucho mayor en los experimentos ejecutados en tanto en cuanto se obtienen soluciones que potencialmente pueden alcanzar el mercado en tiempos más cortos.

Uno de los escenarios ideales en los que implantar las infraestructuras mencionadas es el de las ciudades. Ello permite por un lado disponer de una plataforma sobre la que experimentar, pero a la vez posibilita el utilizar ésta de forma concurrente con objeto de proveer servicio a los ciudadanos y por tanto, involucrando al usuario final en la cadena de valor desde el primer instante. Es así que nace el proyecto SmartSantander piedra angular sobre la que se erige la presente Tesis Doctoral. Para ello, inicialmente se analizan, describen y validan aspectos relativos a las técnicas que posibilitan una gestión uniforme de los dispositivos. Seguidamente, y con objetivo de consolidar la infraestructura mencionada se postula una arquitectura que sea capaz de acomodar tanto la experimentación y la provisión de servicio, todo ello con la consideración de alcanzar un despliegue de más de 12.000 sensores distribuidos en el entorno de una ciudad. En particular, el diseño, implementación y validación de un protocolo de reprogramación remoto de los nodos es una componente esencial para la facilidad experimental que, a su vez, constituye una de las partes nucleares del trabajo. Asimismo, la Tesis incluye sendos capítulos dedicados a mostrar cómo, de la mano del protocolo anterior, se está en condiciones de dar soporte a experimentos variados así como de desplegar y optimizar determinados servicios urbanos.

Finalmente, se concluye presentando las líneas futuras entre las que destacan, la federación de infraestructuras de experimentación y la simbiosis entre el paradigma IoT y las redes sociales.

ABSTRACT

From its birth, the evolution of Wireless Sensor Networks, regarding to their capacity in both processing and storage, as well as in sensing new parameters, has translated into an increase in terms of complexity and heterogeneity associated to them. Between years 2008 and 2009, when the number of devices (things) connected through the Internet already exceeded the number of connected people, it was coined the term of Internet of Things. With the current growth rate, it is estimated that in year 2020 several tens of billions of devices, equipped with different types of sensors and actuators, will be connected to Internet through heterogeneous access networks. In this sense, issues such as the scale, heterogeneity and limitations of the devices, as well as the different nature of the interactions among them, are posed as challenges for a successful integration within the architecture of the Future Internet.

With these boundary conditions, it becomes essential to have experimental infrastructures, within the Internet of Things area, which make it possible to investigate and innovate with the typical technologies associated to these. This type of infrastructures present enormous advantages compared with laboratory environments, or even, with simulation tools, as they are able to work with parameters and considerations based on the State of the Art of the technology, at the same time that operation real conditions and required Quality of Service demands put on top. Furthermore, when that type of experimental facilities bring closer to the end users, it is expected a much higher impact in the experiments carried out, insofar as solutions that can potentially reach the market in shorter times.

One of the ideal scenarios in which introducing the aforementioned infrastructures is represented by the cities. It allows on the one hand having a platform to experiment over it, but at the same time it is possible to use this in a concurrent way, in order to provide citizens with services and therefore, involving end user in the value chain from the first moment. Then SmartSantader was born, becoming the cornerstone over which this Thesis carries out. For this purpose, aspects concerning to the techniques that make it possible a uniform management of the devices, are initially analyzed, described and validated. After this and with the aim of consolidating the mentioned infrastructure, it is posed an architecture able to suit both experimentation and service provision, together with the consideration of reaching a deployment of more than 12,000 sensors distributed in the city environment. In particular, the design, implementation and validation of a protocol for reprogramming nodes in a remote way states as an essential component for the experimental facility that, in turn, constitutes one of the nuclear parts of the work. Additionally, the Thesis includes both chapters addressed to show how, based on previous protocol, it is possible to provide support to varied experiments as well as to deploy and optimize particular urban services.

Finally, it is concluded presenting the future lines among which they can be highlighted, the federation of experimentation infrastructures and the symbiosis between the IoT paradigm and the social networks.

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 INTRODUCCIÓN

Este capítulo presenta el ámbito en el que se sustenta y desarrolla el núcleo de esta Tesis Doctoral. Para ello, partiendo del marco que proporcionan las redes de sensores inalámbricos, se fundamenta la elección de una infraestructura experimental, compuesta por un elevado número de nodos de la Internet de las cosas (superior a 10.000), desplegada en un entorno urbano. Es así que se ha estructurado el trabajo en base a un conjunto de capítulos que abordan los aspectos relativos a la gestión uniforme de sensores heterogéneos (en términos de capacidad de cómputo y comunicación); la descripción de una arquitectura que cubra los requerimientos de los planos de experimentación, gestión y servicio; el diseño, desarrollo y validación de una protocolo de reconfiguración de los nodos constitutivos y cómo éste sirve de base para soportar nuevos experimentos y servicios. Finalmente, se señalan las principales conclusiones así como las líneas de actuación futuras más relevantes.

1.1 INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS

1.1.1 Introducción

El concepto de ciudad inteligente [Schaffers11], [Nam11], y sus dominios de aplicación [Hernández11] están adquiriendo una posición prominente en las tendencias de innovación actuales. En términos generales, la Internet del futuro (*Future Internet, FI*) [SURVEY_FI] y las tecnologías de la información y la comunicación (TIC), consideran la ciudad inteligente como un concepto clave para los futuros desarrollos tecnológicos. En particular, las redes de sensores inalámbricos (*wireless sensor networks, WSNs*), las comunicaciones máquina a máquina (*machine to machine, M2M*) [M2M], se erigen como los habilitadores básicos para cubrir los requerimientos asociados a una ciudad inteligente, dentro del paradigma de la Internet de las cosas (*Internet of Things, IoT*) [Botterman09], [Ashton09].

Las WSNs fueron inicialmente concebidas como redes estáticas ideadas principalmente para la monitorización de parámetros ambientales, en lugares de difícil acceso y sin una arquitectura fija existente. En este sentido, los primeros despliegues realizados se caracterizaban por topologías sencillas de un reducido número de dispositivos, homogéneos entre sí y con baja capacidad de procesado y memoria.

Sin embargo, desde su nacimiento, la evolución de las WSNs ha sido imparable [Akyildiz02], tanto en lo que se refiere a la complejidad de las topologías y el tamaño de los despliegues (redes jerárquicas, movilidad de los nodos), al aumento de la capacidad de procesado y memoria de los nodos, así como a la coexistencia de dispositivos heterogéneos constituyendo las redes desplegadas. El manejo de esta complejidad y heterogeneidad, se traduce en la necesidad de entidades a nivel superior (*middleware*), que permitan la abstracción de las capacidades y características de las redes heterogéneas subyacentes, permitiendo y facilitando el acceso a los dispositivos desplegados de una manera homogénea y uniforme para los usuarios finales.

Adicionalmente al manejo de la heterogeneidad de las interfaces radio subyacentes, el aumento de la capacidad de procesado y memoria, así como de las interfaces de comunicación de los dispositivos que componen las WSNs, ha contribuido a la consolidación de las denominadas comunicaciones M2M; en las que los dispositivos interaccionan de manera directa entre ellos sin necesidad de la intervención humana. Esta capacidad de comunicación convierte a estos dispositivos en potenciales integrantes de la red internet, extendiéndose el enfoque de la misma y surgiendo el concepto de Internet del futuro, la cual debe albergar todos estos dispositivos (cosas), integrándolos y conformando de esta forma la denominada Internet de las cosas. Es por consiguiente una realidad que, tanto las actuales como las futuras tecnologías inalámbricas se encontrarán afectadas por estos despliegues masivos de dispositivos IoT, requiriéndose de manera indispensable, la provisión a la comunidad científica de las infraestructuras que permitan analizar y evaluar los diferentes mecanismos necesarios para la integración de la IoT dentro de la Internet del futuro.

La iniciativa FIRE (*Future Internet Research and Experimentation*) [Gavras07] promovida por la Comisión Europea tiene como principal objetivo la investigación en el ámbito de la Internet del

Futuro, y en particular en la IoT, a partir de la experimentación y validación sobre despliegues a gran escala [Gluhak11].

Las ciudades inteligentes se erigen en un punto de encuentro natural entre la creación de redes experimentales a gran escala (auspiciadas por iniciativas como FIRE), consideradas como excelentes campos de pruebas para la experimentación y la investigación para la FI y el soporte, la participación y el compromiso del usuario con iniciativas innovadoras basadas en las TIC y fomentadas por los denominados laboratorios vivos (*LL, Living Labs*). Éstos se conciben como entornos de experimentación y validación reales, donde los usuarios e investigadores y empresas colaboran hacia el avance en la innovación, habiéndose creado al efecto una red europea (*European Network of Living Labs, ENOLL*) [ENOLL] para la federación internacional de todos estos laboratorios. En este sentido, las WSNs se posicionan como un elemento clave para recoger y generar una gran parte de la información asociada a estos entornos propios de ciudades inteligentes.

Sin embargo, en algunas ocasiones esta visión tan tecnológica de la ciudad inteligente, conlleva la pérdida del objetivo primordial asociado a la misma, el ciudadano. Es fundamental fomentar la participación y el compromiso de la sociedad, orientando las tecnologías más modernas hacia el desempeño de nuevos papeles dentro del diseño, la organización y la planificación de una ciudad, con la finalidad de generar nuevos servicios para el usuario. De esta manera, la ciudad inteligente representa un sistema holístico, en el que considerando los cambios positivos en el comportamiento (asociados a estos nuevos servicios), necesita la involucración del ciudadano desde el primer momento para que éste se sienta partícipe de la modernización de la ciudad. Por lo tanto, desde esta perspectiva, el concepto de ciudad inteligente se asocia con la mejora de la eficiencia de los servicios urbanos, con un desarrollo más sostenible de las ciudades y con la mejora de la calidad de vida de los ciudadanos. La explotación de las diferentes fuentes de información interconectándolas entre sí, así como las capacidades de actuación asociadas a este tipo de despliegues dentro de una ciudad inteligente, permitirán mejorar los actuales servicios urbanos ofrecidos a los ciudadanos.

1.1.2 Motivación y objetivos

Con estas condiciones de contorno el presente trabajo aborda, como objetivo global, las problemáticas asociada a las redes de sensores en dos dimensiones principales. La primera relativa a la gestión uniforme de los nodos, abstrayéndose las características de cómputo, energía y capacidad de comunicación propias de cada uno de ellos. La segunda dimensión pone en valor las metodologías y técnicas anteriores con objeto de posibilitar despliegues masivos como los que se precisan en el ámbito de las plataformas de experimentación a gran escala o de las ciudades inteligentes. Todo ello sustanciado en proyectos y experiencias de carácter nacional e internacional algunas de las cuales se han convertido en casos de éxito de ámbito global.

A fin de abordar el objetivo global mencionado, se han identificado los siguientes objetivos concretos que marcan el itinerario de la presente Tesis Doctoral:

- Concepción, diseño y contribución a la implementación de una arquitectura de alto nivel que permita la provisión de servicio, la capacidad de experimentación y la gestión remota, sobre los dispositivos IoT desplegados dentro del entorno de una ciudad inteligente. Esta arquitectura debe ser lo suficientemente flexible y adaptable para poder incluir nuevos dispositivos, así como

nuevos servicios y aplicaciones de manera sencilla. Además, debe ser una plataforma genérica y replicable, incluyendo elementos fácilmente desarrollables en otros entornos urbanos.

- Definición de los correspondientes módulos y entidades de bajo nivel que realicen las funcionalidades indicadas en la arquitectura de alto nivel, describiendo las correspondientes interfaces de comunicación entre todos estos módulos. De esta forma, se detallarán tanto la arquitectura de dispositivos (referido como *hardware* de ahora en delante) como el logicial (referido como *software* de ahora en delante) específico, para la definición de los bloques constituyentes de esta arquitectura, los cuales se encargarán de proveer las siguientes funcionalidades:
  - La adaptación de diferentes redes de sensores heterogéneas: Mediante la definición, implementación y validación de una capa intermedia (referido como *middleware* de ahora en delante) de adaptación y abstracción se ofrecerá, a las capas superiores, un acceso homogéneo y uniforme a los dispositivos subyacentes. Este middleware debe implementar los correspondientes mecanismos y funcionalidades que permitan recibir información de los nodos, así como acceder a los mismos para gestionarlos de una manera sencilla desde el usuario de las capas superiores.
  - Capacidad de gestión y reprogramación remota. Los nodos desplegados han de permitir la posibilidad de acceder a ellos de manera remota, facilitando la configuración de los mismos mediante el envío de determinados comandos y su correspondiente respuesta. Adicionalmente, con objeto de dotar a la red con la máxima flexibilidad y adaptabilidad, los nodos deben ofrecer la posibilidad de poder ser reprogramado remotamente, permitiendo cambiar el comportamiento de los mismos (la imagen de código que se ejecuta sobre ellos), de acuerdo con las particularidades asociadas a un experimento, así como los requerimientos necesarios de un determinado servicio/aplicación.
  - Soporte a la experimentación. La arquitectura definida deberá permitir el acceso de la comunidad científica a la plataforma para poder realizar los experimentos correspondientes sobre uno o un conjunto de nodos. Para esta finalidad, se deben habilitar los mecanismos para poder seleccionar los nodos y reservar los recursos que intervendrán en el experimento.
  - Provisión de servicio. Cuando los diferentes dispositivos IoT que se encuentran desplegados en el entorno de una ciudad inteligente realicen medidas sobre diferentes parámetros, tales como medioambientales, riego, aparcamiento o tráfico, la arquitectura definida deberá ser capaz de recolectar y almacenar los valores generados por los dispositivos de medida, haciéndolos accesibles a los diferentes proveedores de servicio así como desarrolladores de aplicaciones.
- Funcionalidad y operatividad de la plataforma desplegada. La arquitectura desarrollada debe ser lo suficientemente dinámica y flexible para ofrecer a los diferentes usuarios, de manera concurrente, tanto la capacidad de gestión remota, como el soporte a la experimentación y la provisión de servicio. De esta forma, se asegura la continuidad del servicio así como la capacidad de acceso a los nodos para su gestión remota, independientemente de los experimentos que se estén realizando sobre los mismos. Aunque, como se ha comentado con anterioridad, el concepto de ciudad inteligente se asocia con la mejora de la eficiencia de los servicios urbanos, con un desarrollo más sostenible de las ciudades, y con la mejora de la calidad de vida de los ciudadanos (parte relativa a la provisión de servicio), la capacidad de soportar la experimentación de manera simultánea a la provisión de servicio, permite extender el concepto

de ciudad inteligente convirtiendo a los entornos urbanos en campos de prueba reales de nuevas tecnologías/aplicaciones, que se erijan en la semilla de futuros e innovadores servicios urbanos.

El desarrollo, dentro de este trabajo, de los diferentes objetivos descritos permite la gestión integral de un despliegue masivo IoT en el entorno de una ciudad inteligente, involucrando a un amplio grupo de actores, desde los experimentadores a los proveedores de servicio y desarrolladores de aplicaciones, así como a las Administraciones Públicas y a los ciudadanos. Estos últimos representan a los protagonistas principales para los que se conciben las ciudades inteligentes, incrementando la eficiencia y la sostenibilidad de las mismas a la vez que se mejora la calidad de vida de los ciudadanos.

1.2 ESTRUCTURA DEL DOCUMENTO

En aras de cubrir los diferentes objetivos descritos y de responder a la motivación principal del trabajo, éste se organiza en 7 capítulos.

Tras esta introducción, el Capítulo 2, resuelve la problemática asociada a la gestión de interfaces radio subyacentes heterogéneas mediante el uso de un *middleware* específico, y con la finalidad de uniformizar y homogeneizar el acceso del usuario final a estas interfaces. En este sentido, dentro de este capítulo se indican los principales estándares y tecnologías relacionadas con las redes de sensores, las principales ventajas e inconvenientes asociadas a este tipo de redes, así como las aplicaciones más relevantes para las que éstas se utilizan. En relación al *middleware* citado se han desarrollado y analizado distintas implementaciones, indicando sus principales características y diferencias. Finalmente, a fin de demostrar la operatividad de las implementaciones definidas, se utilizan éstas sobre dos escenarios de aplicación específicos, uno orientado a la monitorización, y el otro dirigido a la gestión de diferentes interfaces radio heterogéneas.

Teniendo en consideración la mencionada evolución de las redes de sensores y la proliferación de despliegues masivos, en el Capítulo 3 se presenta el estado del arte describiendo las diferentes iniciativas y proyectos que, por parte de la comunidad científica y, tanto a nivel nacional como a nivel europeo y mundial, se encuentran llevándose a cabo con el objetivo de profundizar en el estudio de la FI y de los despliegues IoT en el entorno de una ciudad inteligente.

De entre todos los proyectos/iniciativas descritos, cabe destacar el proyecto SmartSantander que persigue un doble objetivo, ofreciendo por un lado un banco de pruebas para la experimentación de nuevas tecnologías y arquitecturas habilitadoras para la IoT, y realizando por el otro, el desarrollo de diferentes servicios y aplicaciones, sobre la arquitectura desplegada, y destinadas a los ciudadanos. En este sentido, el proyecto incluye un despliegue masivo de 20.000 dispositivos IoT, 12.000 de los cuales se instalan en la ciudad de Santander, sirviendo esta infraestructura como marco principal en el que se encuadra y sobe el que se realiza esta Tesis Doctoral.

Con objeto de cubrir esta doble vertiente experimentación-servicio, en el Capítulo 3 se define la arquitectura de alto y bajo nivel con los correspondientes módulos y entidades que la conforman y que, de forma concurrente, soportan la provisión a los servicios a la vez que la dotan de la capacidad de experimentación sobre la infraestructura desplegada (todo ello acompañado de una gestión remota de los diferentes dispositivos desplegados). La arquitectura definida se caracteriza por

presentar un enfoque jerárquico de tres niveles: nodo IoT, pasarela y servidor. De esta forma, gran parte de los nodos (la mayoría caracterizados por una capacidad de cómputo limitada), se agrupan bajo el dominio de una pasarela, la cual se erige como punto intermedio para habilitar las comunicaciones entre el servidor (red dorsal) y los nodos IoT (red capilar), tales como el envío de la información recogida por los dispositivos IoT, así como el tráfico correspondiente a la gestión de la red.

Los Capítulos 4, 5 y 6, conforman otro de los núcleos del presente trabajo, abordando tres pilares básicos de SmartSantander; gestión de la red, provisión de servicio y capacidad de experimentación. Así, el Capítulo 4 presenta el diseño y optimización de un protocolo para la configuración y gestión remota, de la infraestructura desplegada. Atendiendo a las características particulares asociadas a un entorno urbano, interferente y no limitado (en términos de número de dispositivos, ubicación), se describen detalladamente las principales características y el funcionamiento del protocolo desarrollado. Finalmente, a fin de caracterizarlo, así como para demostrar su correcto funcionamiento, se realiza una batería de medidas en diferentes escenarios, tanto en interiores como en exteriores.

En el Capítulo 5, se presenta el diseño, la implementación y la validación de un experimento sobre la arquitectura propuesta en el Capítulo 3. El experimento elegido consiste en un protocolo para el descubrimiento de vecinos en la red, indicándose el funcionamiento específico del mismo, así como los principales detalles de la implementación realizada.

En el Capítulo 6, se aborda el soporte al plano de servicio sobre la arquitectura de SmartSantander. Como un ejemplo ilustrativo, se presenta una mejora sobre el servicio de gestión de plazas de aparcamiento (uno de los casos de uso definido en el Capítulo 3), dotando al mismo de mayor flexibilidad y eficiencia en el proceso de detección. Asimismo, se realiza una comparativa del funcionamiento de ambas aproximaciones, destacando las principales ventajas del nuevo protocolo, tanto en términos de detección, como de gestión de los nodos de aparcamiento, observando también el comportamiento en términos de consumo de batería.

Finalmente, el Capítulo 7 recoge las conclusiones principales del trabajo realizado, así como las líneas futuras que la plataforma SmartSantander ofrece a los diferentes actores involucrados. La experimentación en términos de tecnológicos y orientado a servicios abre una infinidad de posibilidades, pero sin marginar aspectos relativos a la sostenibilidad o al análisis de indicadores clave de prestación de servicio.

2 REDES DE SENSORES AUTÓNOMAS

Las redes de sensores inalámbricos heterogéneas (y de forma más global, las redes de dispositivos IoT) constituyen el pilar sobre el que se erige este trabajo. Así, en este capítulo se identifican las problemáticas más relevantes asociadas a aquéllas y las correspondientes soluciones, basadas en la inclusión de una capa intermedia destinada a abstraer las particularidades de la entidades subyacentes, tales como las interfaces de comunicaciones.

Con objeto de validar la solución adoptada se desarrollan dos aplicaciones, una destinada a dar soporte al traspaso vertical y otra a la monitorización de redes de sensores heterogéneas, que ilustran las ventajas de la aproximación seleccionada.

2.1 INTRODUCCIÓN A LAS REDES DE SENSORES

Aunque la información contextual fue inicialmente considerada relevante para un grupo de servicios pequeño y acotado, actualmente se erige como la base de muchas aplicaciones de diferente índole, incluyendo algunas nuevas surgidas en respuesta a las recientes necesidades de los usuarios, así como servicios ya existentes cuya funcionalidad ha sido extendida (y mejorada), mediante el uso de dicha información contextual. Una de las principales piedras angulares para acceder a este tipo de información son las redes de sensores inalámbricos, las cuales se encargan de capturar y devolver la información correspondiente.

Las WSNs fueron inicialmente concebidas como redes estáticas compuestas por un gran número de dispositivos (nodos sensores), que monitorizaban el valor de varios parámetros, habitualmente de tipo medioambiental, enviando estos valores de manera inalámbrica (utilizando protocolos a varios saltos) a un nodo central, que actuaba como concentrador. Este hecho limitaba la funcionalidad de las redes de sensores inalámbricas a aspectos de monitorización y comunicación sencilla.

Sin embargo, los avances acaecidos en relación al desarrollo y la capacidad de medida, el aumento del número de despliegues de este tipo de redes, así como el creciente interés dentro de la comunidad científica, se ha traducido en un incremento del nivel de complejidad de aquéllas. Esto se refleja en el desarrollo de protocolos de enrutamiento más robustos y eficientes, requeridos para gestionar y manejar cambios de topología de la red, asociados a altas/bajas de un nodo dentro de la red, a la movilidad de los nodos o a la variación en su potencia de transmisión.

Por otro lado, el avance tecnológico se ha traducido en la evolución de los sensores, desde un elemento de medida y baja capacidad de cómputo, hasta un dispositivo con una considerable capacidad de proceso y almacenamiento de información. Ello conlleva un comportamiento más autónomo de las redes de sensores, pudiendo tomar decisiones de manera local (a nivel de nodo), a partir del tratamiento de la información recopilada.

2.1.1 Estándares y tecnologías en redes de sensores

En la actualidad, existen una gran cantidad de tecnologías inalámbricas con sus correspondientes estándares asociados. En principio, según se ha indicado anteriormente, las WSNs estaban asociadas con reducidos consumos y tasas de transferencia bajas, aunque la aparición de dispositivos de mayor capacidad de proceso y memoria, así como una alimentación continua o cuasi-continua del dispositivo, conllevan un nuevo paradigma de las redes de sensores. De esta forma, dentro de las WSNs se pueden englobar a un conjunto de diferentes tecnologías inalámbricas de corto alcance, locales y de área extensa, como se indica a continuación.

2.1.1.1 Redes de corto alcance

- RFID (*Radio frequency identification*)/NFC (*Near Field Communication*) [Want06]: RFID es un término genérico que describe un sistema que transmite la identidad de un objeto o persona de manera inalámbrica. NFC es un subgrupo de RFID que limita el rango de comunicación a unos centímetros. RFID puede trabajar en 4 rangos de frecuencia: baja (9 – 135KHz), alta (13.56MHz), ultra-alta (433 MHz, 868-930 MHz) y microondas (2.45 GHz y 5.8 GHz). Las etiquetas RFID se

pueden clasificar, bien por su capacidad de comunicación (activas y pasivas), o por la posibilidad de escribir sobre ellas (lectura, lectura/una sóla escritura y lectura/ escritura).

- RFID activas y pasivas. Las etiquetas RFID activas utilizan baterías internas para poder enviar la información de manera autónoma. A diferencia de éstas, las etiquetas RFID pasivas utilizan la potencia del lector como fuente de alimentación, de forma que estas etiquetas sólo responderán cuando el correspondiente módulo lector se acerque para leerlas.
- RFID de lectura, lectura/una sola escritura y lectura/escritura. Las etiquetas RFID de lectura se escriben en el momento de su fabricación sin poder ser modificadas posteriormente. Por el contrario, las etiquetas RFID de lectura/una sola escritura, una vez instaladas, pueden escribirse una sola vez, mientras que las de lectura/escritura pueden escribirse tantas veces como sea necesario.

Teniendo en cuenta que actualmente los nuevos dispositivos móviles incluyen lectores NFC, éstos pueden ser utilizados para la lectura de las etiquetas RFID, tanto pasivas como activas. En este sentido, la interacción de estos dispositivos con etiquetas RFID, permite convertirlos en nodos sensores que toman información de estas etiquetas, pudiendo almacenarla, procesarla, e incluso, enviarla a otros dispositivos.

- IEEE 802.15.4 [IEEE802.15.4]: Estándar más utilizado en las redes de sensores, ideado para bajas tasas de transmisión (20 Kbps, 40 Kbps y 250 Kbps) con tiempos de duración de batería muy elevados (con la ayuda de mecanismos de gestión de potencia para disminuir el consumo) y aplicado a dispositivos de baja complejidad. Las frecuencias de trabajo se encuentran dentro de la banda ISM (2.4 GHz, 915 MHz y 868 MHz). Tomando como base este estándar, se han definido diferentes protocolos de gestión de red sobre él, que se detallan a continuación:
  - *Zigbee* [Craig03]: Es un estándar abierto, con lo que permite la interoperabilidad entre dispositivos de distintos fabricantes, así como también provee la capacidad de realizar actualizaciones de firmware inalámbricas. Además, *Zigbee* ofrece la posibilidad de establecer perfiles para aplicaciones comunes como la gestión de energía o el control de la iluminación.
  - *Digimesh* [Digimesh]: Es un protocolo propietario que permite un control más estricto del espacio de código y, por lo tanto, mayor capacidad para ofrecer nuevas funcionalidades. Puede trabajar sobre plataformas con un mayor rango de tasas de transmisión radio que Zigbee. La carga útil de una trama es generalmente mayor, con el consiguiente aumento del caudal eficaz para aplicaciones que tengan que enviar bloques de mayor tamaño. Adicionalmente, incorpora un método de direccionamiento simplificado, que mejora la configuración de la red así como la resolución de problemas de comunicación.
- *Wavenis* [García-Hernando08]: Tecnología desarrollada por la empresa *Coronis* [CORONIS], que provee un gran rango de servicios para dispositivos autónomos extremadamente limitados en términos de batería, ofreciendo altos rangos de conectividad y consumos muy bajos. *Wavenis* extiende el estándar Bluetooth para proveer soluciones inalámbricas e implementar redes utilizando dispositivos autónomos alimentados con batería.
- Entre las principales características de *Wavenis*, se puede destacar que trabaja en las frecuencias ISM de 433, 868 y 915 MHz, presentando una tasa de transferencia máxima de 100 Kbps y capacidad de comunicación a varios saltos entre los nodos que constituyen la red.
- IEEEE 802.15.1 [IEEE802.15.1]: Estándar que está basado y complementa a la especificación Bluetooth v1.1, definiendo las capas de transporte a bajo nivel. De esta forma, Bluetooth que nació como una especificación industrial para comunicaciones radio de corto alcance para

dispositivos personales portátiles queda revisada y adaptada a estándar. Este estándar no es muy utilizado en las WSNs propiamente dichas, si no en la conexión de varios dispositivos (teléfono móvil, videoconsola), en el entorno de un usuario, formando las conocidas como redes de área personal (*Personal Area Networks, PAN*). Bluetooth trabaja en la frecuencia de 2.4 GHZ con tasas de transferencia de 1 Mbps y alcances de 10 m, aunque en su última especificación se han alcanzado valores de 24 Mbps y 100 m de alcance.

2.1.1.2 Redes locales

- IEEE 802.11 [Ferro05]: En principio, relacionada con redes de área local asociadas a un determinado punto de acceso y ofreciendo una conexión de mayor velocidad que los estándares anteriores. Aunque los continuos avances tecnológicos tienden a ubicar interfaces más complejas y de mayor alcance (como 802.11) en los nodos sensores, habitualmente éstos no suelen estar provistos de este tipo de interfaces (implican mayor consumo, precio, complejidad). Sin embargo, para la cobertura de zonas extensas, una de las soluciones radica en utilizar concentradores/pasarelas, que poseen una interfaz de baja velocidad (802.15.4) para recibir los datos de los sensores desplegados en su área de cobertura y otra de media velocidad (802.11) para enviar esta información al resto de concentradores de la red. De esta forma, se crearán dos niveles de red, una a bajo nivel (elementos de medida) y otra a alto nivel (concentradores/pasarelas). Dentro del estándar 802.11, se definen las siguientes especificaciones:
  - 802.11b Extensión del estándar 802.11 para proporcionar tasas de transmisión de datos de hasta 11Mbps usando DSSS (*Direct Sequence Spread Spectrum*) y trabajando en la banda de 2.4Ghz.
  - Wi-Fi (*Wireless Fidelity*) Promulgado por el WECA (*Wireless Ethernet Compatibility Alliance*) para certificar productos 802.11b capaces de interactuar con los de otros fabricantes.
  - 802.11a Extensión de 802.11 para ofrecer velocidades de hasta 54Mbps usando OFDM (*Orthogonal Frequency Division Multiplexing*), trabajando en la banda de 5.4GHz. Esta frecuencia de trabajo, superior a la de 802.11b/g, implica un rango de cobertura más bajo para este tipo de redes.
  - 802.11g Extensión de 802.11 para proporcionar tasas de 20 a 54Mbps usando DSSS y OFDM, también en la banda de 2.4GHz. Es compatible hacia atrás con 802.11b, y proporciona mayor alcance con un menor consumo de potencia que 802.11a.
- IEEE 802.11n [Shrivastava08]: Este estándar define modificaciones tanto en la capa física como en la capa MAC (*Medium Access Control*) 802.11, ya que los modos de operación definidos se optimizan para trabajar con tasas de transferencia mayores (de al menos 100Mbps). Este aumento de velocidad posiciona a los estándares 802.11, no sólo como redes de mayor velocidad que las anteriores, sino también como potenciales redes de respaldo, conjuntamente con las redes cableadas. El estándar 802.11n, trabaja tanto a frecuencias de 2.4 GHz (802.11b/g), así como a 5.4 GHz (802.11a), permitiendo la compatibilidad con los estándares anteriores.
- IEEE 802.11p [Jiang08]: También conocido como WAVE (*Wireless Access in Vehicular Environments*) [Uzcategui09], este grupo se centra en estudiar entornos de comunicación, ya sea entre varios vehículos o entre un vehículo y una determinada baliza fija, donde las propiedades de la capa física subyacente son altamente cambiantes y donde los intercambios de datos se encuentran asociados a una comunicación de corta duración. WAVE trabaja en la frecuencia de 5.9 GHz, con unas velocidades de 18 a 27 Mbps, pudiendo alcanzar distancias de hasta 1000 m.

2.1.1.3 Redes celulares

- GPRS (*General Packet Radio Service*) [Ghribi00]/UMTS (*Universal Mobile Telecommunications System*) [Richardson00]: A diferencia de las tecnologías anteriormente citadas, éstas presentan una cobertura de tipo global, de forma que intrínsecamente no se pueden considerar dentro de una WSN. Sin embargo, al igual que en el caso de 802.11, los módulos GPRS/UMTS pueden actuar como concentradores/pasarelas de un conjunto de nodos sensores. Por ejemplo, se pueden recoger las medidas de un conjunto de sensores (802.15.4) dentro de un vehículo, siendo estos valores enviados a través de un concentrador/pasarela GPRS/UMTS (con cobertura global). Tanto GPRS como UMTS, son redes por las que se transmite tanto tráfico de voz como de datos, aunque dentro de este trabajo nos centraremos en la transmisión de datos. En este sentido, la tasa máxima de transferencia que ofrece GPRS es de 115 Kbps, la cual puede aumentar a 384 Kbps en EGPRS (*Enhanced GPRS*) asociada al protocolo EDGE (*Enhanced Data rates for Global Evolution*), mientras que en UMTS la capacidad máxima, utilizando técnicas de HSPA (*High Speed Packet Access*) puede llegar a los 21 Mbps, cuadriplicándose esa velocidad con el uso de técnicas avanzadas HSPA+.
- LTE (*Long Term Evolution*) [Cox12]: También conocido como 4G LTE, éste es un estándar de alta velocidad para dispositivos de datos y terminales móviles. LTE, basada en capa física, en el estándar OFDMA (*Orthogonal Frequency Division Multiple Access*), permitiendo alcanzar altas tasas de transmisión y elevados volúmenes de datos. La tasa teórica de transmisión más elevada es de 170 Mbps en el enlace de subida y, mediante la utilización de técnicas MIMO (*Multiple Input Multiple Output*), puede alcanzar tasas de 300 Mbps en el enlace de bajada.

En este trabajo nos centraremos principalmente en la tecnología 802.15.4, asociada a despliegues de sensores con alimentación limitada y autónoma, así como con baja capacidad de procesado y memoria.

Hay que reseñar, no obstante que tecnologías como NFC y Bluetooth se pueden desplegar conjuntamente con las tecnologías 802.15.4, constituyendo WSNs híbridas, que permitan una mayor versatilidad en función del entorno y los requerimientos asociados a una determinada situación.

Finalmente, hay que considerar que una red de sensores también puede estar formada por dispositivos en movimiento que necesiten de tecnologías de alcance global (GPRS/UMTS, o en un futuro LTE) o de tecnologías específicas (802.11p), en función del escenario correspondiente.

2.1.2 Ventajas e inconvenientes de las redes de sensores

Las WSNs como su propio nombre indica conjugan las ventajas e inconvenientes de las redes inalámbricas con las particularidades características de las redes de sensores.

Entre las principales ventajas de las WSNs, se pueden señalar las siguientes :

- Movilidad y facilidad de reconfiguración: Las WSNs permiten cambiar dinámicamente las condiciones de medida asociadas a una determinada zona, alternando de manera sencilla la ubicación de los puntos de medida. En este sentido, la evolución y mejora de los protocolos de enrutamiento, permite que la reconfiguración de estas redes se produzca de manera rápida y automática, soportando incluso la coexistencia entre nodos móviles y fijos.

- Simplicidad y rapidez en la instalación: Como condición inherente a cualquier red de índole inalámbrica, el despliegue de una WSN evita la necesidad de tener que realizar cableado, haciendo la instalación más rápida y sencilla. La rapidez en el despliegue permite la instalación en situaciones de emergencia, mientras que la sencillez las hace adecuadas para zonas de difícil acceso, o en las cuales no se dispone de una infraestructura cableada básica.
- Bajo consumo: Las diferentes tecnologías de las WSNs se han de caracterizar por un bajo consumo, puesto que los nodos sensores habitualmente se encuentran aislados y alimentados con una batería. Esto implica que la duración de la misma debe ser lo más larga posible para evitar su reemplazo en cortos períodos de tiempo, así como el correspondiente incremento del coste asociado a la sustitución por unas baterías nuevas.
- Escalabilidad: Las WSNs pueden ser configuradas en una gran variedad de topologías (estrella, anillo, mallada) para satisfacer las necesidades asociadas a una determinada instalación y aplicación específicas. En este sentido, la implementación de protocolos de encaminamiento a varios saltos permite aumentar el número de nodos de una red para cubrir zonas adicionales, o eliminar aquellos que ya no sean necesarios o que no funcionen de manera adecuada.

Por otro lado, entre los principales inconvenientes de las WSNs se pueden destacar los relativos a:

- Caudal eficaz: Las WSNs están ideadas principalmente para el envío de bajas cantidades de datos de manera periódica, junto a diferentes alarmas y eventos que se pueden disparar de forma asíncrona. Este aspecto, unido a los bajos consumos asociados a este tipo de redes, hacen que las tasas de transferencia asociadas a las WSNs sean muy bajas en comparación con otras tecnologías inalámbricas (*WiFi*, *Bluetooth*) y, por supuesto, con las de índole cableado.
- Capacidad de procesado y memoria: Como se ha comentado con anterioridad, las WSNs surgieron como redes para la monitorización de valores ambientales y su correspondiente envío, directamente o tras un procesamiento sencillo, a un concentrador de mayor capacidad de cómputo y memoria. Esta baja capacidad computacional unida a las limitaciones de memoria de los nodos de una WSN, implica que los códigos y aplicaciones que se ejecuten sobre los mismos sean tan sencillas como sea posible. Es importante resaltar que, como se indicó anteriormente, comienzan a proliferar soluciones con nodos más potentes a nivel de procesado y con una mayor capacidad de memoria, haciendo de estos nodos elementos más autónomos y auto-gestionables.
- Alcance: La comunicación entre los elementos de una WSN suele realizarse a través de los diferentes protocolos de encaminamiento a múltiples saltos existentes en la actualidad. Teniendo en consideración las particularidades (colisiones, desvanecimientos) asociadas a un medio de comunicación radio, el número de saltos máximo ha de estar acotado para asegurar una comunicación con una calidad de servicio (*Quality of Service*, *QoS*) adecuada. Esto se traduce en una limitación del tamaño de la red, siendo necesaria la inclusión de los correspondientes concentradores/pasarelas.
- Interferencia: La mayoría de las WSNs trabajan en frecuencias pertenecientes a la banda ISM (*Industrial, Scientific, Medical*), banda reservada para uso no comercial en áreas científica, industrial y médica. Esta banda también es utilizada por otras tecnologías inalámbricas (e.g. *WiFi*), de forma que en lugares de alta concentración de redes inalámbricas trabajando dentro de una misma banda de frecuencias, se pueden producir interferencias.

Hay que tener en consideración que las ventajas e inconvenientes aquí definidos, están asociados a una WSN en general, de forma que se potenciarán unos sobre otros atendiendo a los requerimientos del servicio/aplicación para el que sean desplegadas.

2.1.3 Aplicaciones de las redes de sensores

Al igual que ocurre con las diferentes tecnologías y estándares que han surgido bajo el paraguas de las redes de sensores inalámbricas, el abanico de aplicaciones que utilizan este tipo de redes también ha aumentado y evolucionado. En este sentido, algunas de las aplicaciones más interesantes son las siguientes:

- Monitorización medioambiental: Las redes de sensores nacieron con la finalidad de realizar la monitorización de diferentes parámetros ambientales asociados a una determinada zona. Estas aplicaciones tienen cabida tanto en entornos exteriores (monitorización de campos de cultivo), como en entornos interiores (museos, hospitales). Asimismo, estas medidas se pueden llevar a cabo con nodos estáticos emplazados en determinados puntos estratégicos previamente definidos, así como en vehículos en movimiento para cubrir zonas mucho más amplias.
- Recogida de residuos sostenible: Para evitar la ineficiencia asociada a las recogidas de basura programadas y con una ruta predefinida, sensores de medición de ocupación de contendor (ya sea por peso o por nivel de llenado), permiten la gestión inteligente de las rutas de recogida. En este sentido, se pueden optimizar dichas rutas para evitar recoger contenedores con un bajo nivel de llenado, o para que los camiones de recogida realicen itinerarios que maximicen su capacidad de almacenamiento.
- Detección de vehículos: Mediante la utilización de diferentes tecnologías (ferromagnético, infrarrojo), existen distintas aplicaciones para detectar el estado de ocupación de una plaza de aparcamiento (primero desarrollado en interiores y ahora extensible a exteriores), así como para medir diferentes parámetros asociados al tráfico, tales como la velocidad del vehículo o el volumen de tráfico.
- Alumbrado inteligente: Diferentes aplicaciones para el ahorro de consumo en alumbrado público utilizan detectores de presencia de personas basados en sensores PIR (*Passive Infrared*), o detectores de vehículos basados en sensores de proximidad y efecto *Doppler*, para adaptar la intensidad lumínica en función del flujo de personas/vehículos en cada instante.
- Riego eficiente: En función de determinadas medidas medioambientales a nivel atmosférico (temperatura, presión, humedad, anemómetro, pluviómetro), así como de las condiciones del suelo (temperatura, humedad), se puede gestionar de manera sostenible el riego en una determinada zona verde, evitando la ineficiencia asociada a los riegos programados.

Las aplicaciones anteriormente descritas, constituyen sólo un ejemplo de las más demandadas en la actualidad, aunque la gran cantidad de los parámetros susceptibles de ser medidos, así como las particularidades del escenario, hacen que el espectro de aplicaciones existentes sea muy amplio.

2.1.4 Mercado de las redes de sensores

El imparable crecimiento de las redes de sensores desde su nacimiento, se ha visto reflejado a nivel de mercado, tanto de modo cuantitativo puesto que el número de empresas ha crecido de manera

exponencial, como cualitativo ofreciendo una pléyade de posibilidades en términos de: comunicación (frecuencias de transmisión), capacidad computacional (procesador, memoria), autonomía (gestión de batería, modos de bajo consumo) o medida (diferentes tipos de parámetros medibles). A continuación, se indican alguna de las compañías más representativas a nivel nacional e internacional:

- Libelium [LIBELIUM]: La empresa Libelium se creó como un spin-off de la Universidad de Zaragoza, y se dedica al diseño y la fabricación de tecnología *hardware* para la implementación de WSNs.
  Las principales líneas de investigación y desarrollo de Libelium son *waspmote* y *meshlium*, como dispositivos de comunicación y procesado. Además de ellos, Libelium ha desarrollado placas de medida específicas, cada una de las cuales aglutina un conjunto de sensores asociados a un determinado campo específico. Como ejemplo de estas placas, se pueden destacar la placa de medición de gases, gestión de eventos, ciudades inteligentes, aparcamiento o agricultura.
  Las comunicaciones entre los diferentes dispositivos se realizan a través de los módulos Xbee desarrollados por la empresa *Digi* [DIGI], sobre el protocolo 802.15.4 a las frecuencias de 868 MHz, 915 MHz y 2.4GHz, utilizando el protocolo de enrutamiento propietario *Digimesh* (desarrollado por la empresa *Digi*), para el establecimiento de la red mallada.
- MEMSIC [MEMSIC]: Descendiente de *Crossbow Technology*, y aprovechando la gran experiencia de la misma en redes de sensores inalámbricas, *MEMSIC* está comprometida para hacer el futuro de la tecnología WSN una realidad.
  Para el desarrollo de aplicaciones individuales, MEMSIC presenta un amplio portfolio de productos como las líneas *eKo*, *MICAz*, *TelosB* e *IRIS*, comercializando para todas ellas kits de desarrollo que permiten a los clientes elegir la solución óptima adaptada a sus necesidades. Cada uno de estos productos presenta unas características de transmisión, memoria, procesamiento y capacidad de medida de parámetros diferentes, con la finalidad de adaptarse a los diferentes requerimientos del servicio a ofrecer.
  Por otro lado, MEMSIC ofrece soluciones que conectan el entorno físico con los sistemas de gestión e información empresariales con la finalidad de proveer soluciones de monitorización, automatización y control para un amplio rango de industrias.
- Urbiotica [URBIOTICA]: Urbiotica es una empresa que busca una solución híbrida entre el urbanismo sostenible y la tecnología más innovadora, desarrollando soluciones específicamente diseñadas para su integración armoniosa en el entorno; con sensores y elementos de comunicación muy robustos, preparados contra actos vandálicos, de reducido consumo y de bajo impacto visual.
  Los sensores desarrollados por Urbiotica, principalmente los módulos U-BOX y U-FLAG, transmiten la información recogida al elemento de comunicación más cercano, utilizando una tecnología de radio frecuencia propietaria de baja potencia, a 868 MHz, denominada U-Sense. Además, una plataforma de gestión modular tipo *Cloud*, permite adaptar las aplicaciones en función de los requerimientos de los clientes.
- Advanticsys [ADVANTIC]: Advanticsys es una compañía con experiencia en el campo de las TIC, especializada en redes de sensores inalámbricos (WSN) y otros sistemas de monitorización remota, cuyos principales campos de aplicación son la eficiencia energética, monitorización ambiental y automatización de procesos industriales.

Advanticsys ofrece una gran variedad de dispositivos, como el XM1000 o el CM5000, completamente compatibles con la plataforma *TelosB* y *TmoteSky*. *TmoteSky* fue anteriormente comercializado por *moteiv*, posteriormente por la empresa Sentilla [SENTILLA] y actualmente por Advanticsys.

- TST Sistemas [TST]: TST es una PYME de base tecnológica creada en el año 2007 en el seno del grupo de ingeniería telemática de la Universidad de Cantabria.
  TST busca integrar la tecnología de radiofrecuencia que mejor se ajuste a los requisitos de una determinada aplicación. Para ello, utiliza dos dispositivos base *TSmoTe* y *TSGaTe* con diferentes módulos de expansión para añadir diferentes tecnologías de comunicación, así como diversos tipos de sensores. Por encima de ellos, se erige *TSmarT* como una plataforma modular de comunicaciones inalámbricas que permite la rápida implementación de aplicaciones M2M, monitorización y control remoto.
- Arduino [ARDUINO]: A diferencia de las compañías anteriormente presentadas que ofrecen soluciones comerciales definidas, Arduino presenta una plataforma para realizar prototipos, de código abierto, basada en un *hardware* y un *software* flexible y fácil de utilizar. Está principalmente dirigido a diseñadores y desarrolladores interesados en crear entornos y objetos interactivos.
  Arduino presenta un gran número de placas *hardware* con diferentes entradas analógicas y digitales para la conexión de diferentes sensores. Tanto el lenguaje de programación (basado en *Wiring* [Wiring]), como el entorno de desarrollo de Arduino (basado en *Processing* [Processing]) son de código abierto para el usuario.

Es importante indicar que este estudio anterior se ha tomado una muestra del gran número de compañías que actualmente están proliferando, bajo el paradigma de las WSNs y hacia los despliegues IoT.

2.2 MIDDLEWARE DE ADAPTACIÓN Y ABSTRACCIÓN DE RECURSOS DE DISPOSITIVOS DE MEDIDA HETEROGÉNEOS

Teniendo en cuenta el amplio abanico de tecnologías inalámbricas, muchas de ellas asociadas a las WSNs, así como la continua aparición y desarrollo de las mismas, la comunidad científica comenzó a dirigir sus esfuerzos hacia soluciones que permitieran gestionar, de una manera uniforme y transparente para el usuario, todas estas tecnologías.

IEEE 802.21 [Dutta05] es un grupo de trabajo que centra sus esfuerzos en facilitar y definir la interoperabilidad entre redes heterogéneas, de manera independiente al medio radio. El objetivo de este grupo de trabajo es el desarrollo de una especificación capaz de proveer la inteligencia suficiente a nivel de capa de red, la cual unida con la información provista por otras capas superiores, permita optimizar los traspasos y la interoperabilidad entre redes heterogéneas, incluyendo tanto las de tipo 802 como aquellas no pertenecientes a este estándar. Este marco de trabajo se centra, principalmente, en la provisión de las primitivas necesarias para llevar a cabo el traspaso, definiendo el momento en que debe iniciarse y la forma en que debe prepararse y comportarse el terminal para realizar el mismo.

2.2.1 Proyectos relacionados

En el seno del Grupo de Ingeniería Telemática de la Universidad de Cantabria, el autor de este trabajo ha participado en varios proyectos, tanto de índole nacional como internacional, centrando parte de sus esfuerzos en la implementación de plataformas intermedias genéricas para tecnologías inalámbricas heterogéneas. A continuación se hace una breve descripción de los mismos y del trabajo que se desarrolla en ellos.

- Gollum (*Generic Open Link-Layer API for Unified Media Access*) [Sooriyabandara06]. Proyecto europeo IST-FP6 (*Information Society Technologies – 6th Framework Programme*), cuyo cometido principal fue proporcionar un API (*Application Programming Interface*) genérico para permitir un acceso uniforme al medio físico, independientemente de la tecnología subyacente. La plataforma genérica desarrollada en este proyecto para cubrir esta finalidad se denominó ULLA (*Unified Link Layer API*).
- AN (*Ambient Networks*) [Belgasmi08] y su extensión ANP2 (*Ambient Networks Phase 2*): Proyectos europeos IST-FP6, con el objetivo principal de establecer un plano de control, que pueda incorporarse a las arquitecturas de comunicaciones actuales, y que permita establecer un entorno de comunicaciones fiable, "ocultando" la heterogeneidad de las infraestructuras subyacentes e, incluso, soportando los cambios dinámicos en los requerimientos y las preferencias de los usuarios y servicios. Para este cometido, el manejo de la heterogeneidad de las interfaces radio subyacentes, se realizó mediante la implementación de una plataforma genérica denominada GLL (*Generic Link Layer*).
- MAGNET (*My Personal Adaptive Global NET*) [Prasad10] y MAGNET BEYOND: Proyectos europeos IST-FP6, que se centraron en la problemática que envuelve a las redes de área personal (*PAN, Personal Area Network*). En este sentido, el proyecto desarrolló una arquitectura que garantizase un comportamiento fiable y seguro; concentrado en satisfacer las necesidades del usuario en relación a este tipo de redes, independientemente de la tecnología radio de las interfaces que intervengan en la comunicación. Para la gestión de estas interfaces, la estructura implementada se denominó UCL (*Universal Convergence Layer*).
- m:Ciudad: [mCiudad] Proyecto encuadrado dentro del programa PROFIT (Programa de Fomento de la Investigación Técnica) [PROFIT], cuyo objetivo fue la investigación tecnológica en comunicaciones móviles para posibilitar la movilidad y ubicuidad en la provisión de servicios, contenidos e información, sobre cualquier medio y a cualquier dispositivo. Para cubrir este requisito de independencia, tanto de medio como de dispositivo, se desarrolla una plataforma intermedia para la gestión homogénea de las interfaces subyacentes. Esta plataforma se denominó PLAMIN (Plataforma de Adaptación de Múltiples Interfaces) [Galache08].
- Easy Wireless [EASY_WIRELESS]: El proyecto Easy Wireless se encuadraba dentro del programa ITEA (*Information Technology for European Advancemen*t), y su principal cometido fue el análisis, diseño, implementación y evaluación de distintas técnicas que permitieran derivar una solución en términos de calidad y continuidad del servicio para usuarios móviles, inmersos en un entorno de redes heterogéneas. El objetivo final era asegurar una conectividad permanente, dinámicamente negociada en virtud de los recursos que la red puede ofrecer en cada momento y de las necesidades concretas de los servicios soportados. Dentro del proyecto se desarrolló una arquitectura para traspasos sencillos entre redes inalámbricas, manteniendo la continuidad del

tráfico de datos y del servicio, así como la negociación de la Calidad de Servicio mientras el usuario se desplaza entre diferentes redes.

- MOBILIA (*Mobility concepts for IMT-Advances*) [MOBILIA]: El proyecto MOBILIA se encuadraba dentro del Programa CELTIC (*Cooperation for a European sustained Leadership in Telecommunications*) [CELTIC], centrándose en las recomendaciones y requerimientos de la ITU (*International Telecommunications Union*) IMT (*International Mobile Telecommunications*) para futuros sistemas inalámbricos de comunicación. Entre los requerimientos de partida se podían destacar las velocidades máximas de transferencia de datos (del orden de 100 Mbps para aplicaciones móviles y 1 Gbps para movilidad baja o nómada). Uno de los aspectos más relevantes de la visión IMT-Advanced, es que las futuras redes estarán previsiblemente constituidas por sistemas de acceso con diferentes tecnologías de comunicación radio interoperables entre sí. A este efecto, Mobilia se focalizó en soluciones capaces de manejar adecuadamente dicha heterogeneidad, tomando como tecnología de referencia WiMAX (*Worldwide Interoperability for Microwave Access*) [802.16], debido al papel predominante que, en principio, jugaría en un futuro cercano. El proyecto contempló dos líneas de actuación: la primera centrada en las capas más inferiores (física y MAC), mientras que la segunda se orientó a soluciones de las capas superiores, llevándose a cabo en ambas análisis mediante simulación, y también a través de implementaciones reales y demostraciones.

Hay que destacar que el desarrollo y la implementación (dentro de los proyectos anteriormente citados) de plataformas intermedias para la gestión uniforme y transparente demandada por la heterogeneidad de las interfaces subyacentes, supuso la aparición de un gran abanico de ventajas en múltiples campos; tanto a nivel de investigación con la consiguiente mejora de las prestaciones de estas plataformas, a nivel de usuario con la resolución de necesidades creadas, así como a nivel comercial con la creación de otras nuevas necesidades que surjan de manera paralela al desarrollo de la tecnología.

Por último, la consolidación de la especificación 802.21 como estándar, abre la posibilidad de la actualización (incluyendo la citada especificación) de las implementaciones descritas, con el fin de converger hacia una homogeneización de los procedimientos y, por lo tanto, a una mayor uniformidad en las comunicaciones entre diferentes plataformas.

2.2.2 Implementación

Respecto a los proyectos y las plataformas anteriormente indicadas, una parte de este trabajo se centra principalmente en tres de ellas: ULLA, GLL y PLAMIN (estructura muy similar a la GLL), por lo que sólo se explicarán detalladamente la ULLA y la GLL.

2.2.2.1 ULLA

La ULLA (*Universal Link Layer API*) [Sooriyabandara08], [Galache07] se erige como una abstracción, independiente del sistema operativo, de un conjunto de diferentes interfaces radio subyacentes, para que éstas puedan ser accedidas permitiendo su uso por aplicaciones de capas superiores. Con el término ULLA se puede hacer referencia, tanto a la API como al sistema que va a hacer uso de la misma, lo que significa que la ULLA no sólo es la interfaz que se presenta al programador/usuario para acceder de manera transparente y homogénea a las plataformas subyacentes, sino que también se puede referir a ella como el sistema que hace uso de esta API. En la siguiente figura se muestra

una arquitectura genérica de la ULLA, asociando sus módulos constituyentes a las diferentes capas del modelo TCP/IP [Wettern97].

Entre las principales funcionalidades provistas por esta plataforma destacan las siguientes:

- Resolución de los problemas de interconexión e interoperabilidad asociados a un gran número de tecnologías diferentes, y de los métodos utilizados para el acceso a las interfaces de comunicación, especialmente en el dominio embebido.
- Resolución de los problemas de abstracción y extensibilidad relacionados con diferentes interfaces inalámbricas subyacentes y tecnologías de red.
- Diseño, desarrollo e implementación parcial de una solución para ocultar los problemas asociados a comunicaciones embebidas en las redes, tanto a nivel middleware, como de aplicación. Los resultados serán aplicables a sistemas tales como controles inalámbricos, procesos industriales, aplicaciones automotrices, terminales móviles e inalámbricos.
- Desarrollo de un método de descripción de una interfaz inalámbrica que puede ser utilizado como una herramienta y un marco de trabajo, para la presentación y extensión del soporte del API para actuales y futuras interfaces inalámbricas.

Como se puede observar en la figura 2.1, la ULLA actúa como plataforma intermedia entre una aplicación y los correspondientes controladores de las diferentes interfaces de las que está provisto un determinado dispositivo. A continuación se detallan de manera sencilla todos los módulos que constituyen esta arquitectura genérica.

- Ctrl 1…n (Controlador 1…n): Software dependiente de la tecnología y del fabricante, que permite el acceso y la configuración de una determinada interfaz radio. Esta capa se corresponde en la pila TCP/IP con la de enlace, puesto que se encarga de homogeneizar el acceso a los nodos subyacentes, presentando una interfaz uniforme respecto al nivel de enlace.

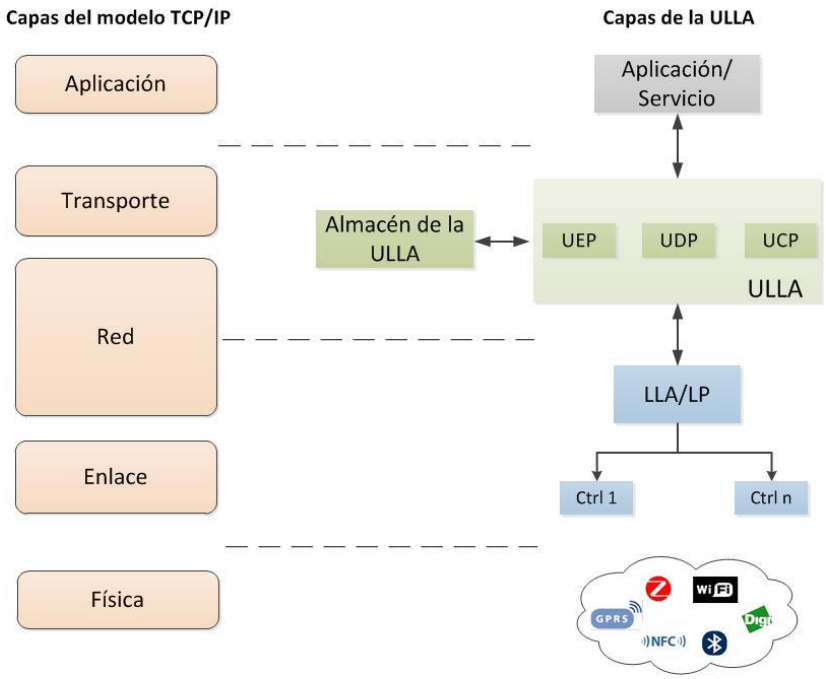

Figura 2.1. Estructura genérica de la ULLA

- Adaptador de Capa de Enlace (*Link Layer Adapter, LLA*): Módulo que se comunica con los diferentes controladores, para recibir la información que necesita de los mismos y reenviársela a la ULLA para que la gestione de manera adecuada. Desde el punto de vista de la pila TCP/IP (Transmission Control Protocol/Internet Protocol), este módulo se encuentra a nivel de la capa de enlace puesto que se encarga de la comunicación con los diferentes controladores, y también a nivel de la capa de red, puesto que se encarga de enviar todos los datos correspondientes a la ULLA que puede encontrarse ubicada en la misma máquina o en una remota.
- Procesador de Eventos de la ULLA (*ULLA Event Processing, UEP*): Gestor de procesos que permite la recepción de los eventos que se produzcan, mayoritariamente en las capas inferiores, con la finalidad de indicar a las capas superiores variaciones de ciertos parámetros radio de alguna de las interfaces radio subyacentes.
- Procesador de Peticiones de la ULLA (*ULLA Query Processing, UQP*): Gestor de peticiones que recibe las solicitudes (por ejemplo, valor de determinados parámetros radio, enlace establecido en ese momento) enviadas por la aplicación, mandándolas al LLA que interactuará con el controlador de la interfaz correspondiente. Una vez el valor es obtenido por el citado controlador, se remite una notificación a la aplicación con el valor requerido.
- Procesador de Comandos de la ULLA (*ULLA Command Processing, UCP*): Gestor de Comandos que resuelve los requerimientos provenientes de la aplicación, destinados a modificar parámetros radio de las distintas interfaces subyacentes, o a configurar las mismas desde las capas superiores.
- Almacén de la ULLA (*ULLA Storage*): Base de datos que permite el almacenamiento de distintos parámetros útiles para el usuario, desde valores de atributos radio propios de las interfaces gestionadas hasta las configuraciones de los enlaces, a través de los cuales se está comunicando el dispositivo.

Los módulos UEP, UQP y UCP, junto con el almacén de la ULLA, se comunican y relacionan entre sí, para conformar la interfaz intermedia entre la aplicación/servicio y el adaptador de capa de enlace, que permite la comunicación con las interfaces radio subyacentes. Desde el punto de vista de la pila TCP/IP, estos módulos se corresponden con la capa de red para su interacción con el LLA y con la capa de transporte, proveyendo la información correspondiente a la capa de Aplicación, en la que se encuentran las diferentes aplicaciones/servicios que utilizarán la ULLA como *middleware* de adaptación.

2.2.2.2 GLL

La GLL (*Generic Link Layer*) [Agüero07], [Pentikousis07], es una plataforma intermedia que provee los medios necesarios para el procesado de datos de capa de enlace de manera universal para múltiples tecnologías de acceso radio. Además, proporciona un conjunto de funciones de capa de enlace que permiten la adaptación e inclusión de nuevas tecnologías de acceso radio. En la figura 2.2 se muestra una estructura genérica de la GLL, asociando sus módulos constituyentes a las diferentes capas del modelo TCP/IP. Como se observa en la ella, la GLL se compone de dos módulos principales:

- Tecnologías de Acceso Radio (*Radio Access Technology, RAT*), representan las diferentes tecnologías radio de las que puede componerse un dispositivo y que se gestionan mediante la GLL. Desde el punto de vista de la pila de protocolos TCP/IP, se corresponde con la capa física,

representando las diferentes interfaces radio heterogéneas subyacentes, así como con la capa de enlace con las funcionalidades necesarias para interaccionar con las interfaces correspondientes.

- Capa de abstracción de la GLL (*GLL Abstraction Layer, gll_AL*): Abstrae las diferentes tecnologías radio subyacentes del gll_IM y, por consiguiente del usuario final, interaccionando directamente con ellas, tomando la información de las capas inferiores (a través de diferentes mecanismos propietarios), y exportando de una manera uniforme la información almacenada. Módulo, cuyos diferentes procedimientos de interacción con las interfaces subyacentes, se basa principalmente en las correspondientes herramientas del sistema operativo. Esta entidad se encuadra dentro de la capa de enlace, homogeneizando el acceso a las interfaces radio subyacentes con la información de cada uno de los módulos inferiores.
- Interfaz y gestión de la GLL (*GLL Interface & Management, gll_IM*): Módulo, cuyo cometido principal radica en ejercer como intermediario con cualquier otro componente que requiera acceder o tomar información de la GLL. En esta capa se encuentra la inteligencia de la GLL, asociada a datos como los parámetros de configuración o los procedimientos de gestión de un enlace. En este sentido, esta entidad se corresponde con las capas de red y de transporte de la pila TCP/IP, permitiendo el acceso de usuarios externos a las diferentes interfaces heterogéneas disponibles, así como haciendo disponible la información recibida desde las capas inferiores.

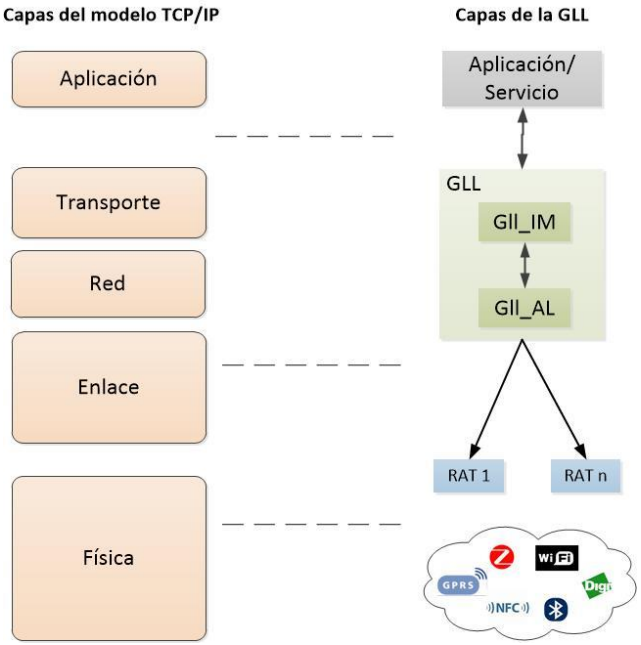

Figura 2.2. Estructura genérica de la GLL

Finalmente, a través de las correspondientes interfaces y métodos implementados por la GLL_IM, las diferentes aplicaciones y servicios a nivel superior pueden interaccionar de manera uniforme y sencilla con las interfaces heterogéneas subyacentes.

2.2.3 Diferencias entre la ULLA y la GLL

Aunque ambas entidades, ULLA y GLL, se caracterizan por presentar soluciones para abstraer de manera homogénea y uniforme las diferentes plataformas y tecnologías radio subyacentes, entre ellas se pueden identificar ciertas diferencias. Una de ellas radica en que, mientras que la ULLA está

ideada para trabajar de manera individual e independiente, actuando como nexo de unión directa entre las aplicaciones y las interfaces de las capas inferiores; la GLL se comporta como un módulo funcional que interactúa con otros, que utilizarán la información provista por la misma, referente a las diferentes interfaces radio subyacentes. De esta forma, gracias a un elemento coordinador, se combinarán todos estos módulos en aras de cubrir una funcionalidad más compleja.

Por otro lado, hay que indicar que en el trabajo desarrollado dentro de la ULLA, se hace una particularización para las redes de sensores inalámbricas, de forma que la estructura generada sea más eficiente y se adapte a las características y limitaciones computacionales de los dispositivos sensores. Esto significa que la citada particularización de la plataforma para redes de sensores se basará en la arquitectura diseñada para la ULLA, utilizando las mismas primitivas de comunicación (comandos, eventos y peticiones), pero definiendo funciones similares más ligeras (en términos computacionales y de memoria), para desarrollar la funcionalidad requerida. A nivel de arquitectura y de comunicación entre módulos, esta plataforma es idéntica a la ULLA; sin embargo hay que tener en cuenta que las limitaciones habitualmente asociadas a los nodos sensores provocan que, a nivel de implementación, haya ciertas diferencias:

- Limitación en términos de memoria y capacidad computacional: Esto se traduce en que la plataforma diseñada sobre estos dispositivos debe ser bastante más ligera que la ULLA, para no consumir memoria en términos excesivos y para que la velocidad del procesador no se ralentice de manera considerable.
- Limitación en términos de batería: Los nodos sensores son dispositivos que difícilmente están alimentados por la corriente eléctrica, de forma que dependen de baterías cuyo tiempo de vida es limitado. Este hecho ha de tenerse en cuenta en el desarrollo del programa que vaya a ser ejecutado sobre los mismos.

2.2.4 Aplicaciones

Como se ha indicado, tanto la GLL/PLAMIN como la ULLA se erigen como entidades emplazadas inmediatamente por encima de los controladores, posibilitando una gestión homogénea de las distintas interfaces subyacentes que potencialmente incorpore un determinado dispositivo.

Sobre las citadas plataformas de gestión se han implementado diferentes aplicaciones, las cuales interactúan con la citada plataforma para obtener la información que necesitan de las interfaces subyacentes, con la finalidad de generar la funcionalidad deseada. Así, se han desarrollado dos aplicaciones, una basada en GLL/PLAMIN ideada para monitorizar y gestionar el traspaso entre redes; y otra implementada sobre la ULLA desarrollando un agente de monitorización de una red de sensores.

2.2.4.1 Monitorización, Gestión y Traspaso entre redes

Esta aplicación está relacionada con escenarios donde coexisten diferentes terminales móviles con varias interfaces de comunicación (802.11 a/b/g), junto con diferentes puntos de acceso emitiendo dentro de las zonas de cobertura de estos terminales. De esta forma, la movilidad de los citados terminales provoca la continua entrada y salida de las zonas de cobertura asociadas a los diferentes puntos de acceso y, por lo tanto, el consiguiente traspaso entre ellas llevado a cabo por el terminal para seguir manteniendo la conexión a la red. En la figura 2.3, se detalla esta situación.

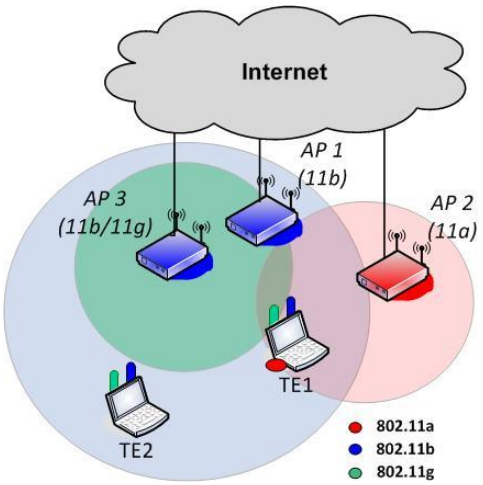

Figura 2.3. Escenario inicial antes del traspaso

Como se puede observar, se presenta un terminal (TE1) provisto de las interfaces 802.11a/b/g y un terminal TE2 con las interfaces 802.11 b/g, así como tres puntos de acceso (*Access Points, APs*) uno emitiendo en 802.11b (AP1), emulado por el terminal TE1, otro en 802.11a (AP2) y un último en 802.11b/g (AP3). El comportamiento del terminal TE1 sería el siguiente:

- Monitorización de las interfaces subyacentes: A través de la funcionalidad provista por la entidad GLL/PLAMIN, se obtiene información acerca de las interfaces radio subyacentes que posee un determinado nodo. La figura 2.4 muestra las interfaces asociadas al terminal TE1.

Interfaces detectadas

	Interfaz	Modo	Tipo	Direccion IP	Direccion MAC	SSID	Canal
1	bge0	ETHERNET	TE	193.144.186.46	00:C0:9F:EB:56:AB	----	----
2	ath0	802.11abg	TE	192.168.2.3	00:14:A4:3E:2C:CB	----	----
3	ath1	802.11b	AP	10.2.2.1	00:13:46:6C:CD:19	AP1	10
4							
5							

Figura 2.4. Interfaces detectadas

Como se observa en la figura 2.4, el terminal TE1 está provisto de tres interfaces, una cableada (Ethernet) denominada bge0 y dos inalámbricas, una de ellas implementando las interfaces 802.11a/b/g (ath0) y la otra actuando como punto de acceso (ath1). Para todas ellas, se indica la dirección IP y MAC, así como el nombre del SSID y el número del canal (para las interfaces inalámbricas).

- Detección de las diferentes redes accesibles. Una vez definidas las interfaces subyacentes que posee el dispositivo, aquéllas que trabajen en modo terminal (en este caso la interfaz ath0), comenzarán a hacer un escaneo del entorno, en busca de las diferentes redes disponibles. En la siguiente figura, se muestra el resultado de este escaneo en el escenario de validación utilizado a tal efecto.

APs detectados

	Interfaz	SSID	Celda	Canal	Modo	RSSI	Condicion
1	ath0	AP1	00:13:46:6C:CD:19	10	Infaestructura	68	No Confiable
2	ath0	AP2	00:11:95:F3:85:2E	36	Infaestructura	65	Confiable
3	ath0	GIT	00:16:B6:2B:84:CE	6	Infaestructura	58	No Confiable
4	ath0	AP3	00:1D:7E:28:20:0D	4	Infaestructura	47	Confiable
5	ath0	GTASGEN	00:16:B6:C1:1E:8F	11	Infaestructura	41	No Confiable
6	ath0	AirGTAS	00:12:17:C2:50:93	4	Infaestructura	40	No Confiable
7	ath0	GTAS	00:13:10:7A:E4:45	7	Infaestructura	24	No Confiable
8							
9							
10							

Figura 2.5. Puntos de Acceso detectados

Como se desprende de la figura 2.5, se han detectado 7 puntos de acceso, de los cuales se ofrece la siguiente información: SSID (*Service Set IDentifier*), celda, canal, modo, RSSI (*Received Signal Strength Indicator*) y condición. Considerando las dos últimas columnas de la tabla, se puede observar como los puntos de acceso están ordenados de mayor a menor relación señal a interferencia (RSSI*)*, y que sólo dos de ellos (AP2 y AP3) son confiables, siendo determinado el grado de confiabilidad de un punto de acceso en función de un fichero de configuración, utilizado para definir los puntos de acceso a los que se puede conectar un determinado nodo. La política de elección del punto de acceso se basa en el valor de la RSSI, con lo que la interfaz ath0 elegirá aquel punto de acceso confiable cuya RSSI sea mayor. En este caso, y como queda claramente señalado en la figura 2.5 (resaltado en color gris), el punto de acceso elegido es el AP2. Este resultado, se encuentra en consonancia con la situación mostrada en la figura 2.3 donde el AP2 se encuentra más cerca del terminal TE1 (en particular que supone una mayor RSSI detectada por el TE1) que el AP3.

- Monitorización de la carga de los puntos de acceso. Para la interfaz del nodo TE1 que actúa como punto de acceso (ath1), se monitorizará su carga, entendida como el número de terminales que se asocian a ella y que queda patente en la figura 2.6.

Datos de carga en los puntos de acceso

	Interfaz	Direccion IP	Direccion MAC	SSID	Canal	Carga
1	ath1	10.2.2.1	00:13:46:6C:CD:19	AP1	10	1
2						
3						
4						
5						

Figura 2.6. Carga asociada a un AP

Se puede observar en la tabla de la figura 2.6 que el dato de carga de este AP es de 1, lo que se corresponde con el escenario de la figura 2.3 y que representa la conexión del terminal TE2.

Enlaces establecidos

	Interfaz	Estado	AP	Canal	RSSI
1	ath0	Conectado	AP2	36	58
2					
3					
4					
5					

Figura 2.7. Calidad del enlace con la red seleccionada

- Asociación a un determinado punto de acceso, mediante la ejecución de un algoritmo de selección. Una vez que la interfaz en modo terminal (ath0) ha elegido el punto de acceso (AP2 en este caso), ésta se conecta con él creándose un enlace entre ambos. En la figura 2.7, se observa la monitorización de la calidad del enlace con la red correspondiente, indicándose las fluctuaciones de RSSI de manera periódica

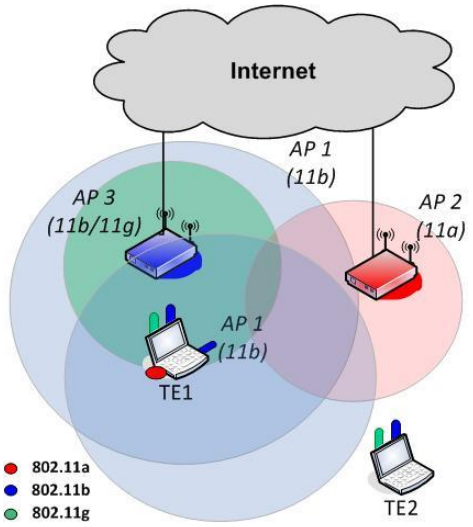

Figura 2.8. Escenario final (después del traspaso)

Una vez establecido el enlace correspondiente, fuerza un desplazamiento del TE1 (hacia la izquierda), indicado en la figura 2.8. El desplazamiento del TE1 provoca la siguiente situación:

- La interfaz en modo terminal detecta un decrecimiento en la RSSI de su enlace con el AP2.
- Mediante un mecanismo de histéresis (implementado en la aplicación de gestión) con unos determinados umbrales máximo y mínimo, se inicia un proceso de escaneo cuando la calidad del enlace está por debajo del umbral mínimo. La razón de haber elegido un mecanismo de histéresis, en lugar de un único umbral fijo, tiene la finalidad de evitar desencadenar continuos procesos de búsqueda cuando el valor de RSSI se encontrase oscilando en torno a este umbral fijo.
- Al realizarse de nuevo el escaneo de la red, se detecta el punto de acceso AP3 como confiable y de mejor RSSI (superior al del AP2), como se recoge de la figura 2.9.

APs detectados

	Interfaz	SSID	Celda	Canal	Modo	RSSI	Condicion
1	ath0	AP3	00:1D:7E:28:20:0D	4	Infaestructura	62	Confiable
2	ath0	GIT	00:16:B6:2B:84:CE	6	Infaestructura	58	No Confiable
3	ath0	AP1	00:13:46:6C:CD:19	10	Infaestructura	56	No Confiable
4	ath0	AP2	00:11:95:F3:85:2E	36	Infaestructura	48	Confiable
5	ath0	AirGTAS	00:12:17:C2:50:93	4	Infaestructura	42	No Confiable
6	ath0	GTASGEN	00:16:B6:C1:1E:8F	11	Infaestructura	32	No Confiable
7	ath0	GTAS	00:13:10:7A:E4:45	7	Infaestructura	20	No Confiable
8							
9							
10							

Figura 2.9. Puntos de Acceso detectados

- El terminal cierra el enlace anterior y establece uno nuevo con el punto de acceso elegido. Se ha producido, por tanto, un traspaso del AP2 al AP3, en este caso, de tipo vertical (entre diferentes tecnologías) de un AP 802.11a a otro 802.11b/g.

Figura 2.10. Calidad del enlace con la red seleccionada

- Por su parte la interfaz que trabaja como punto de acceso (ath1)actualiza su carga, quedándose ahora sin ningún terminal conectado, puesto que el movimiento del TE1 ha supuesto un alejamiento del TE2, quedando éste ahora fuera de la zona de cobertura del AP1.

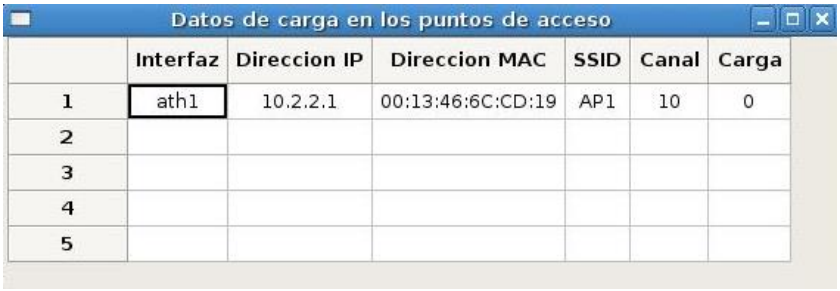

Figura 2.11. Carga asociada a un AP

Como se desprende de las tablas, el módulo diseñado monitoriza las diferentes interfaces, realizando las conexiones/desconexiones (pudiendo definirse parámetros de confiabilidad) a las diferentes redes y gestionando los correspondientes traspasos, en función de la RSSI de los enlaces terminal-punto de acceso.

2.2.4.2 Agente de monitorización

Esta aplicación se ejecuta sobre una red de sensores híbrida, compuesta por dos tipos de nodos diferentes, Mica2 y Micaz, desarrollados por el fabricante MEMSIC. A continuación, se describen las principales características de ambos nodos:

- Micaz [Micaz]: Transceptor radio Chipcon CC2420 compatible con la pila IEEE 802.15.4 y Zigbee. La frecuencia de trabajo es de 2.4 GHz con una tasa máxima de datos de 250Kbps. El mote Micaz posee un microprocesador ATMEL 7.37 MHz ATMega128L, de 8-bits y baja potencia con una RAM de 4KB, 128KB de memoria para programación, 512 KB de memoria *flash* de datos, y 4KB de EEPROM.
- Mica2 [Mica2]: Transceptor radio Chipcon CC1000, trabajando en la banda de frecuencia de 433 MHz (no es 802.15.4/Zigbee). El mote Mica2 posee un microprocesador ATMEL 7.37 MHz ATMega128L, de 8-bits y baja potencia con una RAM de 4KB, 128KB de memoria para programación, 512 KB de memoria *flash* de datos, y 4KB de EEPROM.

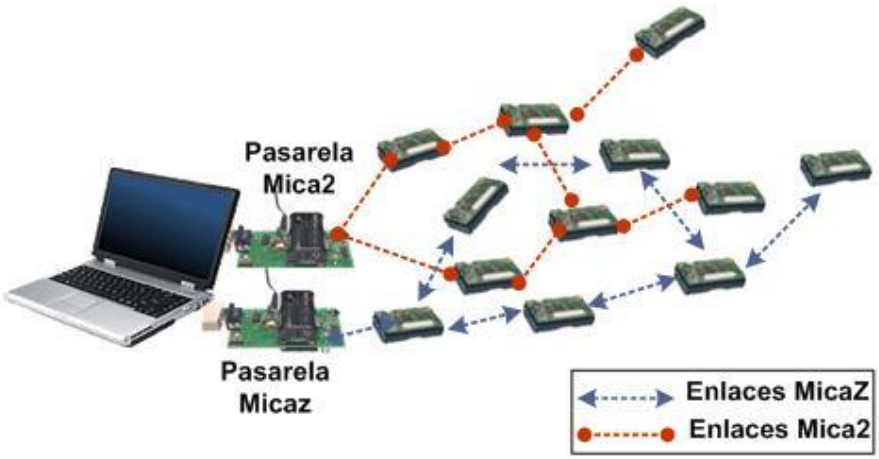

Figura 2.12. Escenario del Agente de Monitorización

La figura 2.12 muestra el escenario de medida, donde se observan dos pasarelas, una de tipo Mica2 y otra MIcaz, para la comunicación con cada una de las redes Mica2 y Micaz, respectivamente. Las redes Mica2 y Micaz trabajan en diferentes frecuencias por lo que son completamente independientes, de forma que las pasarelas correspondientes se encargan de la transmisión/recepción de mensajes entre las redes desplegadas y el ordenador, al que se encuentran conectados a través de un puerto serie.

Lógicamente, las extensiones de la ULLA se instalan en el ordenador, para gestionar tanto el nodo Mica2 como el Micaz, así como en todos los nodos de la red (versión más ligera de la ULLA). La figura 2.13 muestra la funcionalidad de la ULLA para la gestión de ambas plataformas.

Los nodos sensores desplegados, tanto los Mica2 como los Micaz, están provistos de sensores de luz y temperatura, ofreciendo también los datos de batería y de calidad del enlace de comunicaciones. El agente de monitorización (a través de la ULLA), pide todos estos valores mediante sentencias UQL (*ULLA Query Language*), derivado de SQL (*Structured Query Language*) [SQL], mostrándose algunos ejemplos a continuación:

SELECT linkQuality, id FROM Link WHERE lp_id=2

Esta sentencia pide los valores de calidad de enlace y el identificador de los nodos en la red, con la condición de que sólo responda el nodo con identificador 2.

SELECT light, temp, id FROM sensorMeter WHERE temp > 20

En este caso, son luz y temperatura (parámetros ambientales), los atributos cuyos valores son requeridos. Además, se incluye una condición sobre el valor de la temperatura por lo que sólo serán devueltos los valores de luz y temperatura de los nodos que cumplan la condición de la notificación (temperatura > 20ºC). Con la finalidad de conocer los nodos que están cumpliendo esta condición, también es requerido el valor del identificador del nodo.

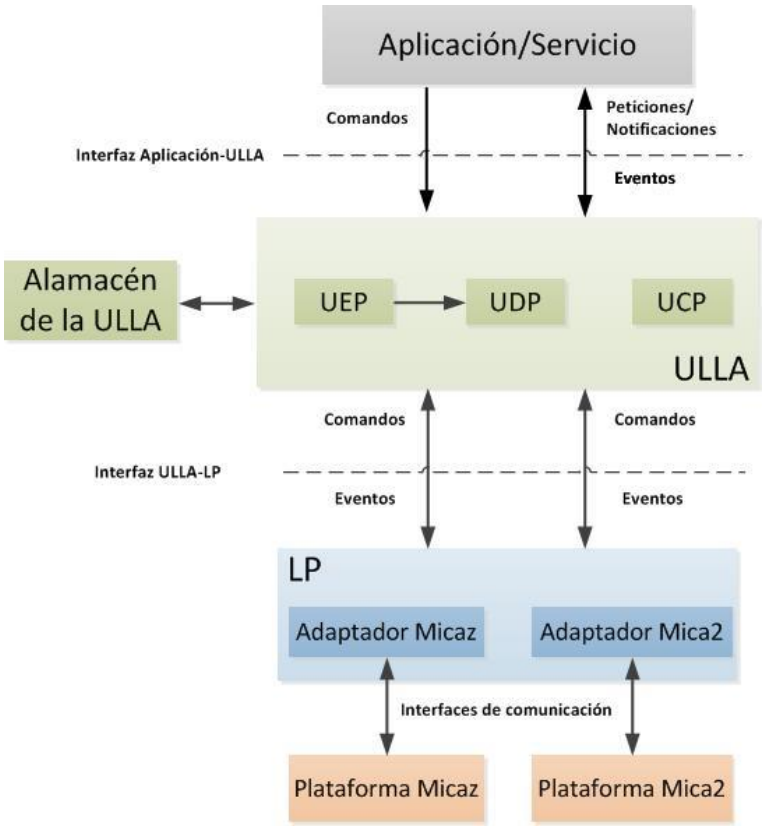

Figura 2.13. Adaptación de las plataformas Mica2 y Micaz a través de la ULLA

Además de los parámetros devueltos y las diferentes condiciones a aplicar sobre los mismos, se ha de especificar el número de muestras que van a ser devueltas y el intervalo temporal entre cada una de las mismas. Lógicamente, los procesos que requieran de un conocimiento muy específico y continuado de la red (por ejemplo en aplicaciones hospitalarias), solicitarán una alta cantidad de muestras con un intervalo pequeño entre ellas. Sin embargo, para aplicaciones no tan exigentes (por ejemplo la medida de la temperatura en un edificio), se realizarán monitorizaciones de valores con un mayor intervalo entre las mismas.

Para el envío de estas peticiones a los nodos, la sentencia UQL debe ser transformada y traducida (por la aplicación ejecutándose en el ordenador) a la estructura de paquetes correspondiente. Una vez realizada la traducción, los paquetes correspondientes se envían a la pasarela, que los procesa y los reenvía a la red de sensores, los cuales reciben el paquete y lo procesan obteniendo los consiguientes valores requeridos en el mismo y almacenándolos temporalmente en el almacén de la ULLA. Además cada nodo implementa un protocolo de enrutamiento a múltiples saltos sencillo, de manera que si no fuese el destinatario de un paquete, lo reenvía como corresponda.

El agente de monitorización consiste en una aplicación Java que, en la parte de transmisión realiza la traducción de las sentencias UQL a los correspondientes paquetes, enviando éstos a la pasarela adecuada (puerto serie correspondiente) y, desde el punto de vista de recepción, se encuentra escuchando al puerto serie del que recibe los correspondientes paquetes con los valores requeridos. Estos valores se envían de manera nativa desde los nodos, es decir según son obtenidos del conversor analógico digital, por lo que son transformados a las unidades correspondientes en las que quieran ser mostrados al usuario (por ejemplo grados centígrados para temperatura, voltios para nivel de batería). A continuación, se muestran dos capturas de pantalla ilustrativas de esta aplicación.

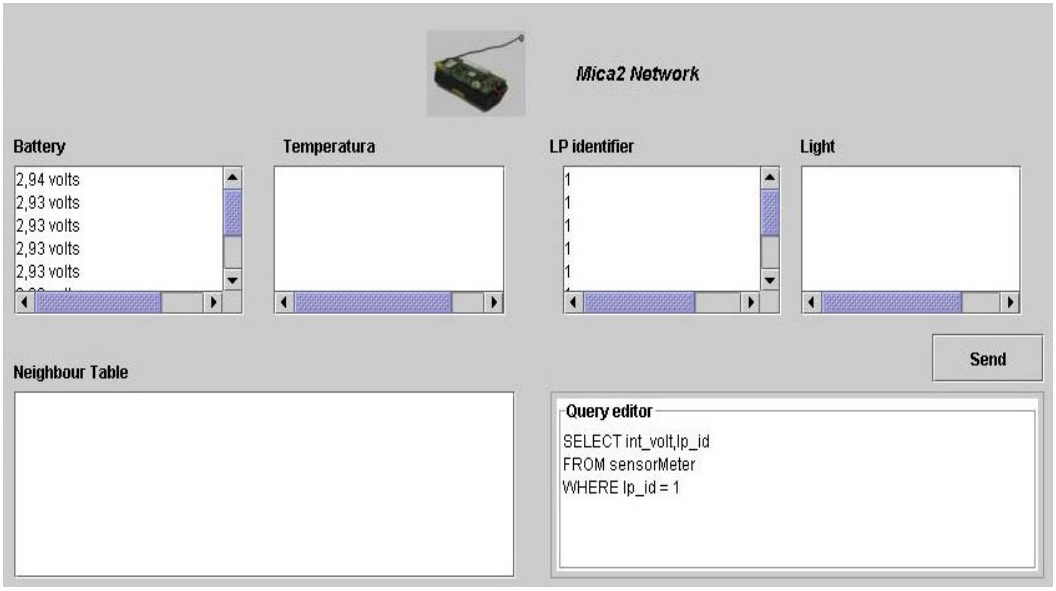

Figura 2.14. Pantalla de monitorización de la red Mica2

La figura 2.14, ilustra los resultados asociados a la red Mica2, donde se muestra la petición del valor de batería (*int_volt*) y el identificador (*lp_id*), con la condición de que sólo sean devueltos estos valores por el nodo con identificador 1. En las columnas superiores se observa que sólo aquellas columnas referentes a la batería y al identificador del nodo se rellenan, y que sólo se reciben valores del nodo 1.

La figura 2.15, ilustra los resultados asociados a la red Micaz, donde se muestra la petición del valor de temperatura (*temp*) y de identificador del nodo (*lp_id*), de los dispositivos cuya temperatura sea mayor de 20ºC. Como se puede observar en las columnas de la parte superior tan sólo se reciben datos de temperatura del nodo 2. En la parte inferior derecha, se presenta un cuadro en el que se editan las diferentes sentencias UQL que se envían a la red de sensores pulsando el botón *Send*. En la parte superior, aparecen varias columnas donde se muestran los diferentes valores de los atributos requeridos a la red de sensores (batería, temperatura, luz e identificador del nodo).

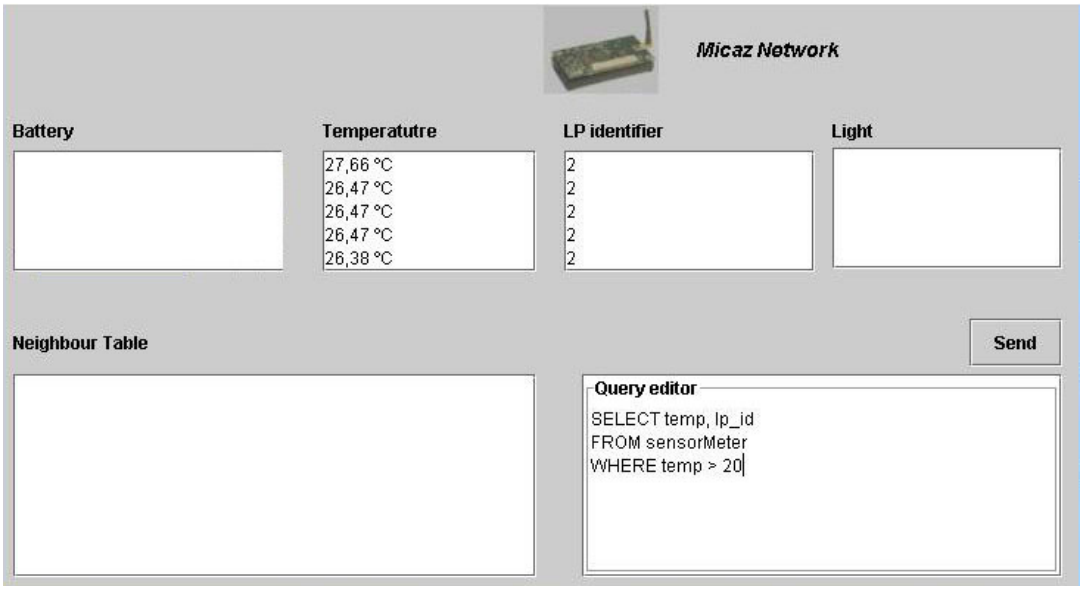

Figura 2.15. Pantalla de monitorización de la red Micaz

Es importante mencionar que estas dos aplicaciones, tanto la remota (ejecutándose en el ordenador), como la local (instalada en los nodos), constituyen tan sólo ejemplos ilustrativos para evaluar algunos de los beneficios que puede proporcionar la API genérica implementada. Sin embargo, su potencial es bastante más extenso, pues una vez recogida la información de la red de sensores en el ordenador, se podría desarrollar una interfaz web, para poder consultar los parámetros monitorizados. Por otro lado, estos valores almacenados también podrían ser utilizados para generar estadísticas, crear históricos de los parámetros monitorizados o desarrollar aplicaciones de valor añadido.

2.3 CONCLUSIONES

Dentro de este capítulo, se ha analizado y resuelto la problemática asociada al acceso, interacción y gestión de múltiples interfaces radio subyacentes heterogéneas. Para ello, se ha presentado el concepto de *middleware* de adaptación y abstracción, que permite uniformizar y homogeneizar el acceso del usuario final a estas interfaces. De manera específica, se han implementado y analizado distintos desarrollos de este tipo de *middleware* (encuadrados dentro de diferentes proyectos nacionales y europeos), indicando sus principales características y diferencias, haciendo especial hincapié en las implementaciones de la ULLA, la GLL y la PLAMIN.

Finalmente, en aras de demostrar la operatividad de las implementaciones seleccionadas, se han utilizado las mismas sobre dos escenarios de aplicación específicos, por un lado para la monitorización y, por el otro, de manera complementaria a la misma, para la gestión de diferentes interfaces radio heterogéneas.

3 INFRAESTRUCTURA DE EXPERIMENTACIÓN EN LA INTERNET DEL FUTURO EN EL CONTEXTO DE LAS CIUDADES INTELIGENTES

En este capítulo, se presenta un amplio estado del arte en relación a las diferentes iniciativas y proyectos, a nivel nacional e internacional, destinadas a avanzar en los campos de la Internet del futuro y la Internet de las cosas, apoyadas en el despliegue de infraestructuras masivas en diferentes entornos, y que dan lugar a los denominados laboratorios vivos.

De entre todos los proyectos/iniciativas descritos cabe destacar SmartSantander, el cual se erige en otro de los pilares de este trabajo, especificando la arquitectura global que ofrezca soporte, de manera concurrente, a la provisión de servicio (a través de la definición de diferentes casos de uso), a la capacidad de experimentación y a la gestión remota, sobre los dispositivos desplegados dentro de un entorno urbano.

3.1 INTRODUCCIÓN

El imparable crecimiento de las redes de sensores inalámbricas, ha significado la consolidación de las comunicaciones máquina a máquina, nacidas como habilitadoras para el flujo de datos entre máquinas sobre una determinada red, y finalmente, a través de una pasarela a un usuario final. Por su parte, el desarrollo de las comunicaciones M2M ha contribuido a la aparición de conceptos como ciudad inteligente (*smart city*) y la Internet de las cosas, como dos de los pilares esenciales sobre los que se erige la Internet del futuro [Botterman09].

La red Internet, nacida en los años 70, se ha convertido en la actualidad, en una infraestructura de comunicaciones y una plataforma de servicios global. Sin embargo, el propósito con el que se concibió dista bastante de los escenarios de uso actuales y futuros, donde nuevas capas y tecnologías de comunicaciones han expandido el concepto inicial de la Internet, con el fin de afrontar los nuevos retos económicos, sociales y tecnológicos que se presentan en la actualidad. En aras de afrontarlos, entre los años 2008 y 2009, cuando el número de dispositivos (cosas) conectados a través de internet excedía el número de personas conectadas, se acuñó el término de la Internet de las cosas [Ashton09]. En esta progresión de crecimiento, para el año 2020 se estima que varias decenas de miles de millones de dispositivos, equipados con diferentes tipos de sensores y actuadores, se encontrarán conectados a internet mediante redes de acceso heterogéneas. En realidad, se puede considerar que la IoT se corresponde con una red de redes con características únicas y particulares, asociadas a cada una de estas redes que la constituyen. En este sentido, la escalabilidad, heterogeneidad y limitaciones de los dispositivos IoT, así como la distinta naturaleza de las interacciones entre ellos, se plantean como retos para una integración exitosa dentro de la arquitectura de la Internet del futuro. El paradigma de la IoT, además, está deparando nuevos retos a la comunidad científica que deben ser tratados adecuadamente para conocer su impacto en el rendimiento de las redes existentes y de las futuras.

El concepto y los dominios de aplicación de la ciudad inteligente están tomando una posición prominente en las actuales tendencias de innovación. En este sentido, la Internet del futuro y, en general, las TIC consideran la ciudad inteligente como un concepto clave para los futuros desarrollos tecnológicos. En particular, las WSN y las comunicaciones M2M se constituyen como los habilitadores básicos para satisfacer los requerimientos derivados del concepto de ciudad inteligente, a la vez que ésta se erige en un excelente entorno de pruebas para la experimentación.

En esta misma línea, se han de considerar los laboratorios vivos, ideados como entornos de experimentación y validación de campo, donde usuarios y empresas de servicios y aplicaciones, pueden crear diferentes innovaciones, generando una gran variedad de experiencias sobre el compromiso del usuario en la innovación basada en las TIC.

En definitiva, la conjunción de las diferentes tecnologías ofrecidas junto con la interacción de los usuarios finales, desemboca en la mejora de los servicios ofrecidos actualmente, así como la generación de otros nuevos de valor añadido.

Desde hace un tiempo, están proliferando, a nivel global, diversas iniciativas dirigidas a avanzar en los campos de la Internet del futuro y la Internet de las cosas, de la mano entre otros de diferentes proyectos de investigación, principalmente enfocados al despliegue de infraestructuras involucrando

actores adicionales a la comunidad científica, tales como ciudadanos, proveedores de servicio y Administraciones Públicas.

3.1.1 Acciones transversales a nivel europeo

Desde su nacimiento en 1984 como Programa Marco 1 (FP1), estos programas se han erigido en una de las fuentes de financiación creadas por la Comisión Europea para la inversión en investigación y desarrollo. Actualmente, este programa se encuentra en su séptima edición FP7 [FP7], que desde 2007 hasta 2013 se encarga de la financiación de diversas áreas de conocimiento.

Nacida en 2008, la FIA (*Future Internet Assembly*) [FIA], es la principal iniciativa europea destinada a fomentar la cooperación entre proyectos con la necesidad de fortalecer actividades a nivel europeo en la Internet del futuro para mantener la competitividad europea en el mercado global. La FIA está destinada a facilitar interacciones entre diferentes dominios técnicos, con el fin de apoyar a la comunidad investigadora dentro de la Internet del futuro a nivel europeo. Actualmente, la FIA engloba un conjunto de alrededor de 150 proyectos de investigación que forman parte del *Challenge 1* del programa ICT (*Information and Communication Technologies*) del FP7, contribuyendo al avance del estado del arte en diferentes iniciativas, todas ellas englobando diferentes proyectos de investigación, así como distintas acciones transversales de soporte a los citados proyectos. Considerando el ámbito en el que se engloba este trabajo, se pueden destacar las siguientes iniciativas: Internet del futuro, FIRE y los laboratorios vivos.

3.1.1.1 Internet del futuro

Para la coordinación de los diferentes objetivos perseguidos por la Internet del futuro, se creó la acción de soporte transversal EIFFEL (*Evolved Internet Future for European Leadership*) [Mähönen06]. La iniciativa EIFFEL busca movilizar a la comunidad investigadora europea para discutir y debatir sobre la Internet del futuro en el desarrollo de la futura sociedad interconectada. Con este propósito, se ha creado un foro de discusión técnica a nivel paneuropeo con el fin de albergar todo tipo de discusiones científicas, así como de determinar las tendencias que siga la sociedad respecto a las redes del futuro. En este sentido, el objetivo general consiste en proveer un ámbito de discusión e intercambio de ideas y trayectorias de investigación en el futuro de la arquitectura de Internet, así como la construcción de los métodos de control correspondiente como fundamento de la sociedad conectada del futuro.

FI-WARE [FI-WARE] se erige como la piedra angular del programa FI-PPP (*Future Internet Public-Private Partnership Programme*) [FI-PPP], con el objetivo de generar una infraestructura de experimentación e innovación, basada en los denominados facilitadores genéricos, que ofrecen funciones reutilizables y compartidas que permiten desarrollar de manera sencilla aplicaciones en múltiples sectores, construyendo de esta forma un fundamento sólido para la Internet del futuro. En este sentido, el proyecto FI-WARE desarrollará especificaciones de los facilitadores genéricos, junto con una implementación de referencia de los mismos destinados a validación y testeo. Además, este trabajo también persigue el objetivo de desarrollar especificaciones de trabajo que servirán como influencia para la definición de los estándares asociados a la Internet del futuro.

3.1.1.2 FIRE

La iniciativa europea FIRE, tiene como objetivo la experimentación y validación en redes a gran escala, mediante la creación de un entorno de investigación multidisciplinar que permite validar experimentalmente e investigar ideas revolucionarias e innovadoras, para nuevos paradigmas de servicio y de interconexión de redes. En este sentido, tres son los principales objetivos perseguidos por esta iniciativa:

- FIRE promueve el concepto de investigación impulsada por la experimentación, combinando la investigación académica con la experimentación y validación a gran escala requerida por parte de la industria.
- Persigue la consolidación de una red experimental, dinámica y sostenible a gran escala y a nivel europeo, que se construya gradualmente mediante la conexión y la federación de bancos de pruebas existentes y futuros, asociados a las tecnologías de la Internet del futuro.
- Finalmente, presenta un marco de trabajo en el cuál la investigación europea en la Internet del futuro pueda florecer, erigiendo a Europa como un actor principal en la definición de conceptos de la Internet del futuro a nivel global.

En la iniciativa FIRE, se incluye la acción de soporte transversal FIRESTATION (*FIRE Support Action*) [FIRESTATION]. FIRESTATION es una acción de soporte que provee aquélla de un concentrador activo destinado a armonizar, guiar y coordinar la demanda y la oferta de las facilidades de experimentación en el contexto de las redes y servicios futuros. El heterogéneo y modular campo de trabajo asociado a FIRE, con grupos de actores de índole nacional e internacional, requiere la compartición de información, la construcción de una comunidad y un punto de contacto único para coordinar y promover la iniciativa respecto a los siguientes requerimientos principales:

- Las redes desplegadas necesitan de la optimización de recursos y esfuerzos comunes para ofrecer a los usuarios el mejor servicio posible, así como para asegurar su sostenibilidad más allá del tiempo de vida del proyecto.
- Los investigadores necesitan un conocimiento adecuado y actual de los recursos disponibles, un acceso sencillo, una alta usabilidad y las herramientas apropiadas para ejecutar y monitorizar sus experimentos.

3.1.1.3 De laboratorios vivos a ciudades inteligentes

En estos momentos que las aplicaciones e infraestructuras de red basadas en la Internet del futuro se encuentran tecnológicamente en primera línea y que, potencialmente podrían suponer beneficios a nivel tanto social como económico, no sólo a la comunidad científica sino también a las ciudades/Administraciones Públicas; el reforzamiento del papel de las ciudades para especificar sus necesidades y requerimientos futuros desde la perspectiva de la innovación dirigida hacia el usuario, se convierte en un aspecto de importancia capital. La identificación de las citadas necesidades y requerimientos realimenta a todas las actividades de investigación, experimentación y despliegue relacionadas con la Internet del futuro, y a los correspondientes bancos de prueba asociados, contribuyendo al establecimiento de diálogo y cooperación entre las diferentes comunidades para formar consorcios entre ellas, en aras de evaluar los beneficios económicos y sociales en las etapas iniciales.

Para esta iniciativa, existe la acción de soporte transversal FIREBALL [Komninos11] y su continuación AMPLIFIRE (*Amplifying Future Internet Research and Experimentation for a Sustainable Future*) [AMPLIFIRE], que establece un mecanismo de coordinación a través del cual un conjunto de ciudades europeas se compromete a una colaboración a largo plazo para adoptar una innovación dirigida al usuario, con la finalidad de explorar las oportunidades de la Internet del futuro. El proceso de coordinación se fundamenta en el intercambio, diálogo y aprendizaje entre ciudades, consideradas como la parte demandante de habilitadores de la innovación en la Internet del futuro. Por otro lado, también se basara en la estrecha relación entre la Internet del futuro, los laboratorios vivos y las ciudades inteligentes.

Los principales objetivos de FIREBALL son los siguientes:

- Alcanzar una coordinación a nivel europeo de las metodologías y propuestas en los dominios de FIRE y los laboratorios vivos.
- Hacer uso de los recursos disponibles en toda Europa para explotar las oportunidades de la Internet del futuro.
- Asegurar el desarrollo coordinado y la compartición de buenas prácticas en la innovación de la Internet del futuro en ciudades y sectores piloto.

La Red de laboratorios vivos (*European Network of Living Labs, ENOLL*) [ENOLL], se asimila a una federación europea de laboratorios vivos referenciados a nivel europeo, incluyendo también otros a nivel mundial, y estando actualmente compuesta por 300 laboratorios vivos.

Desde el punto de vista de infraestructuras de investigación, la iniciativa OneLab [OneLab] provee una facilidad experimental sostenible, a gran escala, compartida, abierta y de propósito general, que permite al sector industrial y académico europeo innovar y evaluar el funcionamiento de sus soluciones. Onelab se basa en los resultados de diferentes proyectos europeos, tales como OpenLab [Tranoris12], NOVI (*Networking innovations Over Virtualized Infrastructures*) [Lymberopoulos12], nacionales como F-LAB [F-LAB], e incluso colaboraciones Unión Europea con otros países como FIBRE (*Future Internet testbeds/experimentation between BRazil and Europe*) [Sallent12], colaboración entre Brasil y la Unión Europea. La federación entre redes, entendida como la conexión entre infraestructuras que permite la compartición mutua de sus recursos, es uno de los objetivos de OneLab, el cual está trabajando en la federación y adición de nuevas infraestructuras, como las desarrolladas en los proyectos G-LAB [G-LAB], FEDERICA (*Federated E-infrastructure Dedicated to European ResearchersInnovating in Computing network Architectures*) [Szegedi10] y ANA (*Autonomic Network Architecture*) [Tschudin07].

A nivel de cooperación entre proyectos, la iniciativa europea IERC (*IoT European Research Cluster*) [IERC], se establece como un grupo de investigación en el marco de la Internet de las cosas, con la finalidad de reunir los diferentes proyectos financiados por la Unión Europea, así como de definir una visión común en los retos de investigación en el desarrollo y la tecnología IoT a nivel europeo. El propósito fundamental relacionado con la Internet de las cosas consiste en abordar el amplio potencial de las capacidades basadas en IoT en Europa – coordinar/fomentar la convergencia del trabajo venidero en los aspectos más importantes – para construir un consenso amplio sobre las directrices para acometer la gestión de la Internet de las cosas en Europa.

De cara al futuro, y como continuación del FP7, el siguiente programa no se denominará FP8 sino Horizonte 2020 [HOR2020], y discurrirá entre los años 2014 y 2020, contribuyendo al impulso para crear nuevo crecimiento y empleo dentro de Europa.

3.1.1.4 Proyectos a nivel europeo

Al amparo de los distintos retos del subprograma ICT del FP7 [ICT-FP7], muchos son los proyectos que se han financiado en el ámbito de la Internet del futuro, mostrándose algunos de los más relevantes en la siguiente figura.

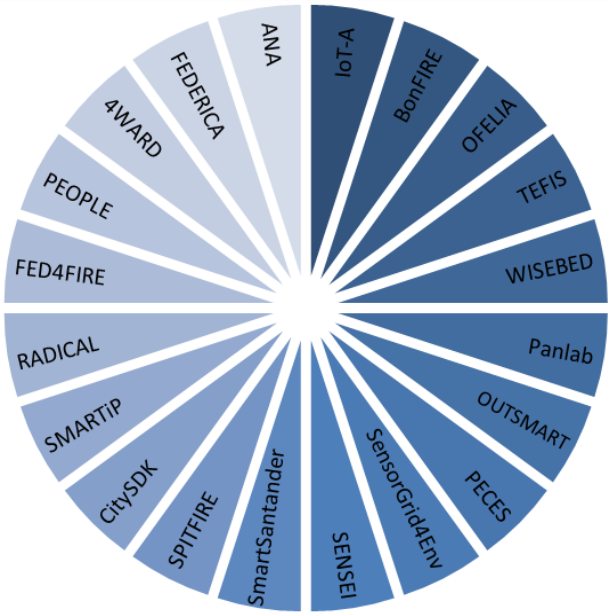

Figura 3.1. Proyectos europeos

El nacimiento de las redes de sensores masivas está principalmente asociado a la experimentación en entornos reales de aplicaciones, servicios, protocolos que antes quedaban limitados al ámbito de la simulación. Siguiendo esta línea, algunos proyectos están destinados a la provisión de bancos de pruebas, así como los correspondientes interfaces y módulos, para facilitar y mejorar la experimentación de la comunidad científica sobre este tipo de redes. Este es el caso de los proyectos WISEBED (*Wireless Sensor Network Testbeds*) [Coulson12], OFELIA (*OpenFlow in Europe: Linking Infrastructure and Applications*) [Kopsel11], PanlabII (*Pan European Laboratory Infrastructure Implementation*) [Wahle11] y FED4FIRE (*Federation for Future Internet Research and Experimentation*) [FED4FIRE] cuya finalidad es interconectar (federar) diferentes bancos de prueba, ofreciendo un acceso transparente y uniforme al usuario a todos los recursos disponibles, así como una sencilla escalabilidad de los mismos, añadiendo nuevos nodos y funcionalidades (como la virtualización), a los despliegues correspondientes. Por otro lado, los proyectos FEDERICA y ANA tienen como principal objetivo la creación de infraestructuras y arquitecturas de red escalables y autónomas, que permitan probar nuevas tecnologías de red sobre ellas. Desde el punto de vista *software*, el proyecto PECES (*PErvasive Computing in Embedded Systems*) [Selvarajah10] desarrolla una capa de adaptación que permite la cooperación de forma transparente entre dispositivos embebidos; mientras que el proyecto BonFIRE [Hume12] busca la convergencia entre redes y servicios, mediante la experimentación y validación de escenarios innovadores en la investigación de la Internet de los servicios. (IoS, *Internet of Services*). Como base a muchos de estos proyectos y en

aras de la creación de un modelo de referencia arquitectural, se erige el proyecto IoT-A (*Internet of Things - Architecture*) [IoT-A] definiendo además un conjunto de bloques constitutivos clave.

Teniendo en cuenta que la mayoría de los despliegues se realizan en entornos urbanos, surge la necesidad de involucrar a los ciudadanos en estos despliegues y, en consecuencia, a las correspondientes Administraciones Públicas. Este objetivo implica no sólo familiarizar a los ciudadanos con estos despliegues masivos, sino hacerles partícipes de los mismos, ya sea involucrándose en ellos de forma directa, o bien disfrutando de la mejora de los servicios urbanos ofrecidos gracias a la implantación de estas redes masivas. Surgen de esta forma, proyectos como OUTSMART [OUTSMART] donde se lleva a cabo el desarrollo de servicios sostenibles asociados a diferentes ecosistemas locales, o SensorGrid4Env (Semantic Sensor Grids for Rapid Application Development for Environmental Management) [SensorGrid4Env], que implementa una arquitectura orientada al desarrollo de servicios asociados a la gestión ambiental. Desde la perspectiva del usuario, el proyecto SMARTiP (Smart Metropolitan Areas Realised Through Innovation & People) [SMARTiP] trata de transformar los servicios públicos en función de la visión de los usuarios derivada de los laboratorios vivos, mientras que RADICAL (RApid Deployment and adoption of sustainable socially-aware and intelligent sensing services for emerging smart cities) [RADICAL] busca mejorar los servicios ofrecidos y crear otros nuevos, combinando la información provista por los sensores desplegados con aquella generada en las redes sociales. Con el objetivo de intentar acelerar y facilitar la aceptación de los innovadores servicios desplegados, se erige el proyecto PEOPLE [PEOPLE].

El proyecto SmartSantander [Sánchez11], [Krčo13], se sitúa dentro de los dos grupos de proyectos definidos anteriormente, puesto que persigue un doble objetivo, ofreciendo un banco de pruebas para la experimentación de nuevas tecnologías y arquitecturas habilitadoras para la IoT, así como persiguiendo el desarrollo de diferentes servicios y aplicaciones, sobre la arquitectura desplegada, destinada a los ciudadanos.

Teniendo en consideración, el elevado número de servicios y aplicaciones que se están desarrollando de manera independiente por los diferentes proyectos, surgen iniciativas con la finalidad de facilitar y homogeneizar la implementación de aplicaciones sobre una determinada arquitectura común. En este sentido, el proyecto TEFIS (*Testbed for Future Internet Services*) [TEFIS] ofrece un punto de acceso para que los desarrolladores de *software* y de negocio puedan validar, experimentar y elaborar conocimiento de manera colaborativa. Por su parte, el proyecto 4WARD [Niebert08] ofrece un acceso directo y ubicuo a la información provista por un conjunto de redes desplegadas a nivel europeo, permitiendo un desarrollo de aplicaciones de red más rápido y sencillo. De una forma similar, la arquitectura desarrollada dentro del proyecto SENSEI (*Integrating the Physical with the Digital World of the Network of the Future*) [Presser09], busca abstraer la heterogeneidad de los recursos desplegados, para habilitar la interacción de la información en el mundo real. Por último, los proyectos CitySDK (*City Service Development Kit*) [CitySDK] y SPITFIRE (*Semantic Service Provisioning for the Internet of Things using Future Internet Research by Experimentation*) [Pfisterer11], tienen como principal objetivo la creación y provisión de un conjunto de herramientas, métodos e interfaces para el desarrollo eficiente de aplicaciones y servicios en el contexto de la Internet de las cosas.

Como se ha comentado, además de los proyectos a nivel europeo, también existen actividades llevadas a cabo a nivel nacional, como la mencionada G-LAB que se erige en la iniciativa alemana

para la creación de una amplia red de experimentación e investigación, destinada a la interacción entre las nuevas tecnologías y los requerimientos de las aplicaciones emergentes. Del mismo modo, en Francia la iniciativa F-LAB [F-LAB] trabaja en la consolidación de una facilidad experimental a gran escala compartida, abierta, sostenible y de propósito general que fomente la irrupción de la Internet del futuro en Francia. A nivel español, la iniciativa es.INTERNET [ESINTERNET] se erige en la Plataforma Tecnológica Española de Convergencia hacia Internet del Futuro, proponiendo un enfoque TIC sectorial que abarca desde las redes e infraestructuras TIC que forman la base de la Internet del futuro, hasta las aplicaciones y servicios desarrollados para otros sectores sobre esta infraestructura.

3.1.1.5 Otras iniciativas

Aunque al estudio anterior del estado del arte se ha centrado principalmente en las iniciativas a nivel europeo, hay que tener en consideración que otros proyectos y despliegues encuadrados dentro del marco de la Internet del futuro a través de los laboratorios vivos y de las ciudades inteligentes, proliferan a nivel mundial. Cabe señalar que la Comisión Europea ha firmado diferentes convenios para la colaboración con diversos países en todo el mundo [COL_EUR].

A nivel americano, varias son las iniciativas que se pueden indicar, GENI (*Global Environment for Network Innovations*) [GENI] es un programa colaborativo soportado por la NSF (*National Science Foundation*) estadounidense, destinado a proveer un banco de pruebas global a gran escala para la validación y testeo de arquitecturas de la Internet del futuro. Iniciado en 2005, ha atraído un amplio interés y participación, tanto del entorno académico como del ámbito industrial. GENI se diferencia de los bancos de pruebas habituales en que concibe una facilidad a gran escala de propósito general, sin establecer límites en las arquitecturas de red, los servicios y las aplicaciones a ser evaluadas, así como permitiendo la generación de nuevos diseños para experimentar con usuarios reales bajo condiciones reales. En este sentido, la idea clave es dividir en múltiples porciones la capacidad ofrecida, con la finalidad de la compartición de servicios y la realización conjunta de diferentes experimentos. En colaboración con el anterior, el proyecto ORBIT (*Open-Access Research Testbed for Next-Generation Wireless Networks*) [ORBIT] presenta un banco de pruebas a dos niveles, un emulador de laboratorio así como una campo de pruebas real, diseñado para lograr la reproducibilidad de la experimentación, al mismo tiempo que soporta la evaluación de protocolos y aplicaciones en despliegues del mundo real.

En Asia, se puede distinguir el programa japonés AKARI (*Architecture Design Project for New Generation Network*) [AKARI], cuyo objetivo se basa en llevar a cabo el diseño de una infraestructura de red del futuro basada en una arquitectura simple, abierta y flexible a cambios futuros y funcionalidades nuevas, donde las estructuras físicas y lógicas se encuentran separadas. La iniciativa china CERNET (*China Education and Research Network*) CERNET], desempeña el papel más importante en la investigación de aplicaciones y tecnologías de red avanzadas, siendo la red sobre la que se llevan a cabo muchos de los proyectos a nivel nacional. Por último, el programa coreano KOREN [KOREN], presenta una red de investigación avanzada que soporta el desarrollo y el despliegue de nuevas tecnologías y aplicaciones dentro de las comunidades I+D.

Finalmente, destaca la red PLanetLab (*An open platform for developing, deploying, and accesssing planetary-scale services*) [PlanetLab], como una red global de experimentación que soporta el desarrollo de nuevos servicios de red. Desde su nacimiento en 2003, más de 1000 investigadores de

las instituciones académicas y los laboratorios de investigación industrial más punteros, han utilizado PLanetLab para desarrollar nuevas tecnologías para almacenamiento distribuido, mapeado de red, sistemas punto a punto y procesado de peticiones. Actualmente PLanetLab consiste en 1157 nodos ubicados en 547 lugares repartidos por diferentes lugares de todo el mundo.

3.2 INFRAESTRUCTURA IOT MASIVA PARA LA EXPERIMENTACIÓN Y LA PROVISIÓN DE SERVICIO

3.2.1 Introducción

Como se ha comentado anteriormente, dentro de los proyectos financiados por la Comisión Europea dentro de la iniciativa FIRE, se encuentra el proyecto SmartSantander, en el que se afronta la realización de un despliegue masivo de 20.000 sensores en las ciudades de Belgrado, Guilford, Lübeck y Santander (en esta última 12.000 dispositivos). El despliegue en Santander, debe servir de banco de pruebas para la implementación y validación de los diferentes desarrollos que se afrontan dentro del marco de esta Tesis Doctoral.

El proyecto ofrece una doble oportunidad de explotación: experimentación y provisión de servicio, manteniendo al mismo tiempo la accesibilidad y gestión de la red, como se describe a continuación:

- Experimentación: La comunidad investigadora dispone de un despliegue, que se ofrece como una infraestructura única para el desarrollo de experimentos en un entorno real. El investigador puede acceder a la plataforma, reservar los recursos (nodos) necesarios dentro de la red durante un tiempo determinado, y reprogramarlo para que ejecuten los experimentos correspondientes; tales como por ejemplo, diferentes tipos de protocolos de enrutamiento, técnicas de codificación de red o procesos de minería de datos.
- Provisión de servicio: Diferentes servicios ideados para cubrir las necesidades y requerimientos de los ciudadanos han sido desplegados. A diferencia de los experimentos, en este caso serán los servicios públicos o los proveedores de servicio, los que determinen el conjunto de nodos que deben estar ejecutando un determinado servicio, así como el tiempo de duración del mismo. Servicios gestionados por la ciudad, tales como la gestión del aparcamiento en exteriores, la monitorización medioambiental, el riego inteligente, así como otros más dirigidos a los ciudadanos, como la realidad aumentada o el sensado participativo, son algunos de los servicios que se han llevado a cabo dentro del proyecto.
- Gestión de la red: Los nodos desplegados en la red son gestionados remotamente por los administradores de la red, permitiendo el envío/recepción de comandos/respuestas hacia y desde los nodos desplegados, así como actualizaciones de firmware de los nodos, modificación de los servicios desplegados y carga de diferentes experimentos mediante la programación remota, denominada OTAP (*Over The Air Programing*) ó MOTAP (*Multihop OTAP*), referida a lo largo de este trabajo de manera indistinta con cualquiera de estos dos términos.

Es importante resaltar que la provisión de servicio y la gestión de la red se llevan a cabo de manera ininterrumpida, y de manera simultánea a la ejecución de los diferentes experimentos que se realicen sobre un determinado nodo.

3.2.2 Actores involucrados

La mayoría de los proyectos citados, tanto de índole nacional como internacional, están dirigidos principalmente a la comunidad investigadora, a través de la realización de desarrollos e implementaciones de módulos o definiciones de conceptos más allá del estado del arte actual, que servirán como semilla para aplicaciones y conceptos utilizados en un futuro por los potenciales usuarios finales. Sin embargo, el proyecto SmartSantander, además de llevar a cabo un despliegue masivo que sirva de banco de pruebas para tecnologías punteras, también presenta una facilidad para la provisión de servicios dirigida a los usuarios finales. En este sentido, el abanico de actores involucrados se extiende más allá de la comunidad científica, incluyendo los siguientes:

- Proveedores de servicio y desarrolladores de aplicaciones: Usuarios de alto nivel que utilizan el banco de pruebas desplegado para crear servicios y aplicaciones basadas en los datos obtenidos de la red de sensores desplegada.
- Ciudadanos y Administraciones Públicas: Consumidores de las aplicaciones/servicios desarrollados por los usuarios (proveedores de servicios/desarrolladores de aplicaciones). Los servicios pueden ser utilizados directamente por los usuarios (aplicaciones para *smartphones*), o demandados por las Administraciones Públicas para mejorar los correspondientes servicios municipales, así como para ofrecer otros nuevos que redunden en una mejora del nivel de vida de los ciudadanos.
- Usuarios FIRE y comunidad científica en general: Usuarios a nivel de experimentación, tales como investigadores con la finalidad de desarrollar, validar y optimizar algoritmos, protocolos, implementaciones y aplicaciones novedosas, en un banco de pruebas a gran escala.
- Administradores de la plataforma SmartSantander: Usuarios con privilegios de gestión responsables de configurar el sistema, mediante la creación de cuentas de usuario, el control de los derechos de acceso y la configuración de los nodos desplegados dentro de la red; así como de gestionar la infraestructura *hardware* (actualización dinámica de la red), y las entidades *software* (acceso a los dispositivos instalados) desplegadas.

Es importante resaltar que los diferentes casos de uso desarrollados dentro del proyecto SmartSantander, no se encuentran dirigidos a todos estos actores descritos anteriormente, sino a uno o varios de los mismos.

3.2.3 Despliegue de la infraestructura

Como se ha comentado, el proyecto de SmartSantander, contempla el despliegue de 20.000 sensores, 12.000 de los cuales se instalan en la ciudad de Santander. Es sobre este despliegue llevado a cabo en la ciudad de Santander, donde se enmarca parte del trabajo realizado en esta Tesis Doctoral.

Actualmente, la red desplegada en Santander [RED_SDR] se compone de alrededor de 3.000 dispositivos IEEE802.15.4, 200 dispositivos provistos de módulo GPRS/3G y 2.600 etiquetas conjuntas RFID y código QR (Quick Response), desplegadas tanto en ubicaciones estáticas (farolas, fachadas, paradas de autobuses), como embarcados en vehículos móviles (autobuses, taxis). Considerando un dispositivo IoT como aquel capaz de medir un determinado parámetro, y teniendo en consideración que los dispositivos indicados anteriormente se encuentran provistos de varios sensores, el número de dispositivos IoT actualmente instalado supera los 12.000.

Figura 3.2. Despliegue de dispositivos IoT en Santander

El despliegue de los dispositivos mostrados en la figura 3.2, se lleva a cabo mediante la definición y la implantación de diferentes casos de uso, todos ellos con la finalidad de cumplir la dualidad provisión de servicio-experimentación, perseguida por el proyecto, así como permitiendo el acceso y la gestión de todos estos dispositivos.

3.2.3.1 Soporte a la provisión de servicio

Dentro de este apartado, se describen los diferentes casos de uso desplegados en el proyecto, indicando los datos provistos por cada uno de ellos a la plataforma de SmartSantander.:

- Monitorización medioambiental estática: Alrededor de 2.000 dispositivos instalados (principalmente en el centro de la ciudad), en farolas, fachadas que proveen medidas sobre diferentes parámetros medioambientales, tales como temperatura, CO, ruido, luminosidad.
- Monitorización medioambiental móvil: Para extender la plataforma de experimentación al ámbito de la movilidad, así como el caso de uso de monitorización medioambiental previamente descrito, adicionalmente a la medida de parámetros en ubicaciones estáticas, dispositivos instalados en 150 vehículos de uso público, (autobuses, taxis, vehículos municipales), devuelven diferentes parámetros ambientales, tales como O_3 (ozono), NO_2, CO, partículas en suspensión, de manera periódica y georeferenciada con la posición donde se realiza la medida. Además de estos parámetros ambientales, en ciertos autobuses parámetros medidos por el CAN-Bus (*Controller Area Network vehicle bus standard*), estándar que permite la comunicación de los diferentes dispositivos y sensores dentro de un vehículo, tales como velocidad, posición GPS (*Global Positioning System*), altitud, rumbo, cuentakilómetros, se almacenan en la plataforma de SmartSantander.
- Gestión de aparcamientos en exteriores: Cerca de 400 sensores de aparcamiento (basados en tecnología ferromagnética) enterrados bajo el asfalto, han sido instalados en las principales áreas de aparcamiento del centro de la ciudad, para detectar la disponibilidad de plazas libres en estas zonas.
- Guiado a plazas de aparcamiento libres en exteriores: Alimentados por la información devuelta por los sensores de aparcamiento desplegados en el caso de uso anteriormente descrito, 10

paneles ubicados en las intersecciones de las principales calles de la *zona 30* ((zona del centro de la ciudad en la que la velocidad se encuentra limitada a 30Km/h), se han instalado para el guiado de los conductores hacia las zonas donde haya disponibilidad de plazas de aparcamiento libres.

- Monitorización de la intensidad de tráfico: Alrededor de 60 dispositivos ubicados en las principales entradas de la ciudad de Santander, han sido desplegados para la medida de los principales parámetros asociados al tráfico, tales como velocidad del vehículo ,grado de ocupación de la calzada, volumen de tráfico y longitud de la cola de vehículos.
- Riego inteligente de parques y jardines: En torno a 50 dispositivos se han desplegado en dos zonas verdes de Santander (Parque de Las Llamas, Finca Altamira y Parque de la Marga), para monitorizar parámetros relacionados con el riego, tales como la temperatura y humedad del suelo, pluviómetro, anemómetro, presión, temperatura y humedad atmosférica, en aras de intentar proporcionar un riego más eficiente.

En la figura 3.2, se puede observar un detalle de la instalación llevada a cabo en la zona centro de la ciudad, el área principal donde se lleva a cabo el despliegue. Sin embargo, adicionalmente a este despliegue, se han realizado despliegues de menor tamaño en sendas zonas periféricas de la ciudad, el PCTCAN (Parque Científico y Tecnológico de Cantabria) y el Campus de la Universidad de Cantabria. Mientras que el despliegue llevado a cabo en la zona centro aglutina todos los casos de uso descritos con anterioridad, los despliegues periféricos presentan medidas asociadas exclusivamente a la monitorización medioambiental, pero se erigen como fundamentales para la realización de pruebas y validaciones de uso en campo real, previa a su implantación en el despliegue del centro de la ciudad.

- Realidad Aumentada: Alrededor de 2.600 etiquetas RFID/código QR conjuntas, se han desplegado con el fin de ofrecer la posibilidad de etiquetar diferentes puntos de interés dentro de la ciudad, tales como lugares públicos (parques, plazas, etc.), puntos de interés turístico, tiendas. A pequeña escala, el servicio provee la oportunidad de distribuir información dentro del entorno urbano como aquella basada en la localización, en función del número de lecturas realizadas sobre una determinada etiqueta.
- Sensado participativo: En este escenario, los usuarios utilizan sus teléfonos móviles para enviar información asociada a los propios sensores del teléfono (dependiendo del modelo y fabricante), tales como las coordenadas GPS, la brújula, la temperatura o el ruido, alimentando esta información a la plataforma de SmartSantander. Los usuarios también pueden subscribirse a servicios como “el pulso de la ciudad”, donde pueden recibir alertas relativas a eventos que ocurran en la ciudad, y a los que se hayan subscrito. También pueden reportar la ocurrencia de ciertos eventos, que serán recibidos por otros usuarios subscritos al evento correspondiente.

Tanto para el caso de la realidad aumentada, como para el caso del sensado participativo, se han desarrollado las correspondientes aplicaciones [Gutiérrez13] para dispositivos móviles en los sistemas operativos Android [Android] e IOS (*iPhone Operating System*) [IOS].

Como se desprende de los casos de uso descritos, además de ofrecer una serie de servicios, los usuarios contribuyen a la generación de una gran cantidad y variedad de datos, los cuales pueden ser utilizados por diferentes entidades en el desarrollo de distintas aplicaciones y servicios.

3.2.3.2 Soporte a la experimentación

Además del correspondiente servicio provisto por los casos de uso anteriormente descritos, los nodos desplegados asociados a cada uno de ellos deben ofrecer la posibilidad de experimentar sobre ellos a agentes externos, ya sean experimentadores, proveedores de servicio o desarrolladores de aplicaciones. En este sentido, dentro del proyecto se definen dos tipos de experimentación:

- Experimentación a nivel de nodo: La mayoría de los dispositivos IoT pueden ser programados remotamente, tantas veces y con tantos programas diferentes como sean requeridos por los experimentadores. La reprogramación de los nodos se lleva a cabo de manera inalámbrica mediante, permitiéndose la reprogramación a múltiples saltos mediante el protocolo MOTAP (detallado en el Capítulo 4). Este tipo de experimentación permite a los investigadores implementar y validar diferentes experimentos, tales como distintos protocolos de enrutamiento, técnicas de codificación de red o algoritmos de minería de datos. Ello es posible de la mano de un transceptor IEEE802.15.4 adicional, de forma que se aísla el tráfico asociado a la experimentación de los datos generados en la provisión del servicio. Algunos de los nodos desplegados, debido a restricciones en consumo de batería, capacidad computacional, tamaño de memoria, o bien por las características de diseño asociadas al fabricante, ni están provistos de un transceptor 802.15,4 adicional ni pueden ser reprogramados inalámbricamente para cargar diferentes experimentos.
- Experimentación a nivel de servicio: Los datos generados por los diferentes nodos desplegados en la plataforma, son ofrecidos a los investigadores, proveedores de servicios o desarrolladores de aplicaciones para mejorar los servicios ya completados o implementar otros servicios y aplicaciones. En este sentido, la creación de servicios de valor añadido, así como la correlación de información ofrecida por diferentes casos de uso, podrían indicarse como ejemplos de este tipo de experimentación.

Es importante reseñar que, mientras que la experimentación a nivel de servicio está disponible en todos los casos de uso, la experimentación a nivel de nodo sólo es ofrecida por algunos de ellos: riego de parques y jardines y monitorización medioambiental estática y móvil. Será en estos últimos casos de uso, en aquellos sobre los que se centra el trabajo desarrollado en este Tesis Doctoral.

Por último, aunque en el apartado anterior se señalaron los despliegues periféricos como bancos de prueba previos a la instalación en el despliegue del centro de la ciudad, desde el punto de vista de experimentación, también se ha realizado un despliegue en interiores (edificio de I+D+i de Ingeniería de Telecomunicación) a menor escala mimetizando el despliegue en exteriores. En este despliegue, se puede acceder a los nodos de manera cableada, permitiendo depurar código y estresar la red de una manera más rápida y eficiente. Lógicamente, dentro de este banco de pruebas también se llevan a cabo pruebas referentes al comportamiento de los nodos respecto a la provisión de los correspondientes servicios a ser desplegados, previa a la carga de los mismos en el despliegue en exteriores.

3.2.3.3 Soporte a la gestión de la red

En aras de garantizar la correcta provisión de servicio así como el adecuado soporte a la experimentación, se debe implementar una eficiente gestión de la red, con las siguientes funcionalidades:

- Envío y recepción de comandos hacia y desde los nodos desplegados, para poder acceder a los parámetros básicos de funcionamiento de los mismos, ya sea a nivel radio (potencia de tx, frecuencia de comunicaciones), así como a nivel de aplicación.
- Manejo de las dos interfaces radio independientes (cuando las haya), tanto a nivel del nodo, donde se ha de diferenciar la interfaz de envío de datos de experimentación con aquella asociada a los datos de servicio y gestión de la red; como a nivel de la pasarela, donde los datos recibidos han de ser correctamente procesados y enviados hacia el correspondiente experimentador, proveedor de servicio, desarrollador de aplicaciones o gestor de la red.
- Programación remota de los nodos de manera inalámbrica a un salto y a varios saltos, de manera *unicast*, *multicast* y *broadcast*, pudiéndose enviar tantos programas como sean necesarios, los cuales se almacenarán en una tarjeta de memoria externa provista por el nodo.
- Todos los dispositivos desplegados deben tener preinstalado un código por defecto (denominado *Golden Image*), que asegure que el nodo se comporte de una manera confiable y accesible. De esta forma, cuando se carga un código erróneo en el nodo, la ejecución de esta imagen permite devolver al nodo a sus valores por defecto y, por lo tanto, a todas las funcionalidades de gestión anteriormente descritas.

El tráfico derivado de la gestión de la red se envía a través de la misma interfaz física que los datos derivados de la provisión de servicio, mientras todo el tráfico asociado a los diferentes experimentos que se realicen sobre los nodos (si está disponible la experimentación a nivel de nodo), se lleva a cabo a través de una interfaz física diferente (transceptor IEEE802.15.4 adicional).

3.2.4 Consideraciones *hardware* de la infraestructura

Tanto este apartado de instalación hardware como el de arquitectura software se centrarán, como se indicó anteriormente, en los casos de uso que permiten la experimentación a nivel de nodo. Para esta finalidad, los nodos poseen dos interfaces radio físicas diferentes, una para la provisión de servicio/gestión de la red y otra para la experimentación, permitiendo por tanto la experimentación a nivel de nodo. Estos casos de uso son el riego inteligente de parques y jardines y la monitorización medioambiental estática y móvil.

3.2.4.1 Nodos estáticos

La monitorización medioambiental estática y el riego inteligente de parques y jardines comparten la misma arquitectura hardware, provista por la compañía española Libelium [LIBELIUM], y que se compone de los siguientes componentes:

- Repetidor: Responsable de medir los parámetros correspondientes a cada caso de uso, tales como temperatura, CO, ruido, luz, presión, temperatura y humedad del aire. Estos nodos se encuentran ubicados en altura, en farolas y fachadas, comportándose como nodos emisores de información, así como nodos reenviadores de la información asociada a otros repetidores. La siguiente figura muestra en detalle el *hardware* instalado.

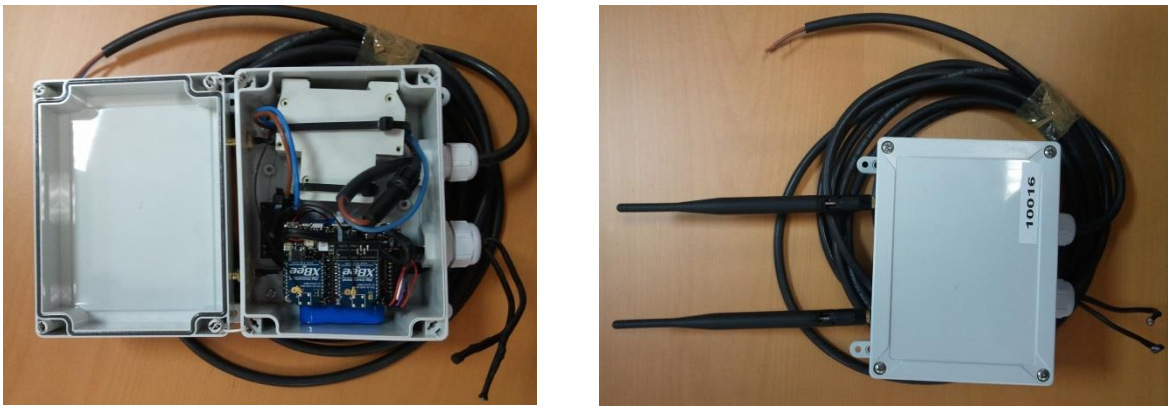

Figura 3.3. Repetidor

El nodo repetidor se compone de las siguientes partes:

- Placa Base: Esta placa (denominada comercialmente *waspmote*) se encarga del procesado de datos y la gestión de la memoria, ofreciendo un microprocesador Atmega1281 y varios tipos de memoria como una SDRAM (*Synchronous dynamic random access memory*) de 8KB, una *flash* de 128KB y una memoria SD (*Secure Digital*) de almacenamiento extra de 2GB de capacidad. Por otro lado, 7 entradas analógicas y 8 digitales, permiten la conexión de sensores externos; unidas a las cuales se encuentran 1 interfaz PWM (*Pulse Width Modulation*), 2 UART (*Universal Asynchronous Receiver/Transmitter*), 1 I2C (*Inter-Integrated Circuit*) y 1 USB(*Universal Serial Bus*), para la conexión de diferentes módulos de comunicaciones.
  - Dos módulos radio XBee-PRO: Ambos módulos radio, fabricados por la compañía *Digi*, trabajan a la frecuencia ISM de 2.4GHz. Uno de ellos implementa el protocolo 802.15.4 en modo nativo, mientras que el otro implementa un protocolo 802.15.4 modificado con la inclusión de un protocolo de enrutamiento propietario (denominado *Digimesh*). Este protocolo de enrutamiento punto a punto, permite un direccionamiento sencillo entre los nodos que forman la red, simplificando la inclusión de nuevos nodos en la red de manera transparente a las capas superiores.
  - Placas de medida: La compañía Libelium ofrece un amplio portfolio de placas de medida dirigidas a diferentes sectores y compuestas de un conjunto de sensores que miden parámetros asociados a estos sectores específicos. Todas estas placas son modulables, de forma que se pueden incluir sólo aquellos sensores que se necesiten, en función de los requerimientos del caso de uso correspondiente. En este caso, se han utilizado las siguientes placas:
    - Placa de gases: Utilizada para el caso de monitorización medioambiental estática, incluida en aquellos repetidores que realizan medidas de temperatura y CO.
    - Placa de ciudades inteligentes: Permite la media de los parámetros de luminosidad y ruido, dentro del caso de uso de monitorización medioambiental estática.
    - Placa de agricultura: Utilizada dentro del caso de uso de riego inteligente de parques y jardines, permite medir los valores de humedad, temperatura y presión atmosférica, parámetros más específicos como pluviómetro, anemómetro y veleta, así como parámetros referentes al estado del terreno, como la humedad y temperatura del suelo.

Figura 3.4. Pasarela (meshlium)

- Pasarelas: La pasarela (denominada comercialmente *meshlium*) se erige como el punto intermedios entre la red capilar de medida (compuesta por los repetidores) y la plataforma central de SmartSantander. En este sentido, al igual que los repetidores, las pasarelas están provistas de la interfaz *Digimesh* para recibir las mencionadas tramas de servicio, así como para el envío de comandos relacionados con la gestión de la red. De la misma forma, las pasarelas, también poseen una interfaz 802.15.4 nativa para comunicarse a nivel de experimentación con los nodo desplegados. Todos los repetidores desplegados se configuran para enviar (a través de la interfaz *Digimesh*) toda la información asociada a la provisión de servicio, ya sea generada por ellos o reenviada de otros nodos, hacia las pasarelas. Una vez que la información es recibida en las pasarelas, ésta se almacena en una base de datos que, o bien puede estar ubicada en un servidor web para permitir el acceso a la misma a través de internet, o bien puede ser enviada a otra máquina (servidor central), a través de las diferentes interfaces provistas por la pasarela (WiFi, GPRS/UMTS y *Ethernet*).

Teniendo en cuenta el tamaño del despliegue, en términos tanto de superficie a cubrir como de número de dispositivos, éste debe organizarse en diferentes *clusters*, conjuntos de nodos en los que la pasarela se erige como nodo concentrador de los diferentes dispositivos que se encuentran desplegados bajo su dominio. Así, se ha de dimensionar la red asegurando que el número de nodos dentro de un *cluster*, así como el número de saltos máximos para acceder del nodo más alejado a su correspondiente pasarela, sean tales que permitan un funcionamiento óptimo de la red. Para evitar que se produzcan interferencias entre *clusters* adyacentes, éstos se configuran en diferentes canales (el módulo XBee en su versión PRO habilita 12 canales en la frecuencia de 2.4 GHz) de forma que. A nivel *Digimesh*, nodos pertenecientes a diferentes *clusters* no podrán comunicarse entre sí directamente, sino a través de sus correspondientes pasarelas. Como se indicará más adelante, por encima de las pasarelas se erigirá un módulo que permita el almacenamiento de los datos provenientes de las mismas, así como la comunicación de la plataforma SmartSantander con todas las pasarelas desplegadas. En la siguiente figura, se muestra un ejemplo de la arquitectura de uno de los mencionados *clusters*.

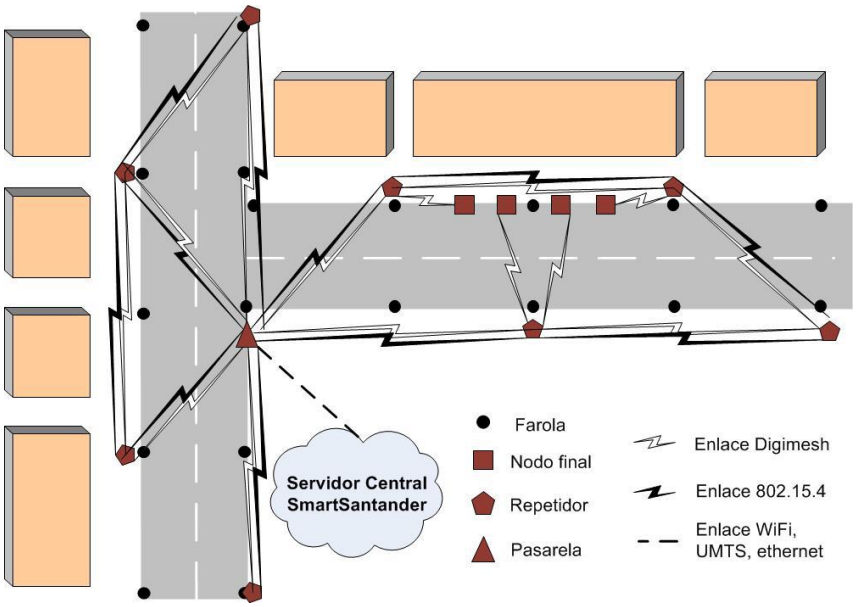

Figura 3.5. Arquitectura hardware del despliegue de nodos fijos

Como se puede observar en la figura 3.5, los diferentes repetidores se encuentran ubicados en las farolas/fachadas comunicándose a varios saltos de distancia con la pasarela a través de la interfaz *Digimesh*. Adicionalmente, tanto los repetidores como la pasarela se encuentran también provistos de una interfaz 802.15.4 nativa para experimentación. Como se ha comentado con anterioridad, esta interfaz nativa carece de un protocolo de enrutamiento, por lo que salvo que este se implemente sobre ella, la comunicación sobre esta interfaz se limita a los nodos vecinos. En este sentido, para asegurar la transmisión de los resultados del experimento a la pasarela y, por consiguiente, al servidor central de SmartSantander, éstos son enviados a través de la interfaz *Digimesh*. Este funcionamiento se explicará de manera más detallada en el Capítulo 5.

Además de los repetidores y las pasarelas, en la figura también se incluyen los nodos finales. Estos nodos se diferencian de los repetidores en i) su ausencia de capacidad para reenviar información (sólo envían y reciben información), ii) suelen estar alimentados por baterías no recargables, con las consiguientes restricciones de consumo, iii) sólo están provistos de la interfaz *Digimesh* para el envío de datos de servicio, no permitiendo realizar experimentación sobre los mismos. Un ejemplo de este tipo de nodos, está representado por los destinados a la detección de ocupación de plazas de aparcamiento, los cuales sólo envían información sobre el estado de las plazas y reciben comandos de configuración. Aunque este trabajo se centra en aquellos casos de uso que permiten la experimentación a nivel de nodo, la ilustración permite conocer de manera global la arquitectura a la que responde el despliegue realizado en la ciudad.

Figura 3.6. Instalación de repetidores en farolas

En la figura 3.6, se observan distintos detalles de la instalación de varios repetidores en diferentes farolas del centro de Santander. Los repetidores se alimentan mediante baterías recargables, de forma que durante la noche (cuando la farola está encendida), las baterías se recargan, mientras que durante el día, los nodos trabajan de manera autónoma. De esta forma, los consumos diurnos buscan quedar compensados por las cargas nocturnas, dependiendo la compensación de carga de batería del nodo del uso de una o de las dos interfaces de comunicación, del número de transmisiones/recepciones de paquetes, así como lógicamente de la duración de los correspondientes ciclos nocturnos y diurnos.

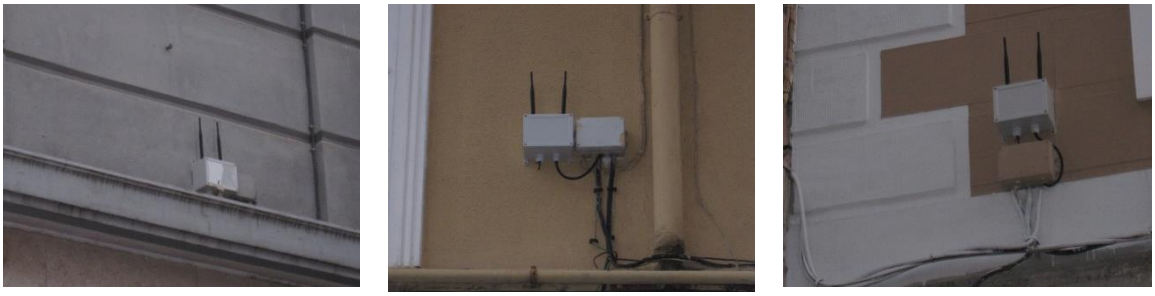

Figura 3.7. Instalación de repetidores en fachadas

En la figura 3.7, se observa un detalle de la instalación de varios repetidores en diferentes fachadas del centro de la ciudad. La instalación en fachadas se lleva a cabo en la mayoría de las calles de la *zona 30*, en las que las luminarias se encuentran ubicadas en las fachadas (no hay postes de farola). Al depender de las luminarias de las fachadas, los ciclos de carga/descarga son los mismos que los de los repetidores ubicados en las farolas.

Figura 3.8. Instalación de un sensor suelo y un repetidor de riego

En la figura 3.8, se observa el detalle de la instalación de un sensor enterrado, para la medida de la humedad y temperatura del suelo, así como el detalle de un repetidor de riego (similar a los repetidores ambientales), que se ubican en las farolas. En este caso, el sensor de suelo se conecta de manera cableada a los repetidores ubicados en farolas.

La figura 3.9 muestra la instalación de diferentes pasarelas, tanto en fachadas como en farolas, en diferentes partes de la ciudad. A diferencia de lo que ocurre con los repetidores, los meshliums necesitan alimentación continúa, mediante PoE (*Power over Ethernet*) para funcionar correctamente,

de forma que tanto en las fachadas (conectándolo a una toma de corriente interna), como en las farolas (aprovechando las que incorporan cámaras dado que requieren de alimentación continúa), las pasarelas se encuentran alimentados de manera continua.

Figura 3.9. Instalación de pasarelas ***en fatolas y fachadas***

En la figura 3.10, se muestra el detalle de un *cluster* de monitorización medioambiental situado en el centro de la ciudad, donde la pasarela (*meshlium*) situada en la parte central inferior, se encarga de gestionar el conjunto de nodos que forman el *cluster*; en este caso, nodos de monitorización de luminosidad y de ruido.

Figura 3.10. Detalle de un cluster de monitorización medioambiental en el centro de la ciudad

En la figura 3.11, se observa el *cluster* de riego ubicado en el parque de Las Llamas, con la correspondiente pasarela para almacenar y procesar todos los datos devueltos por los diferentes dispositivos desplegados. En este sentido, tres tipos diferentes de dispositivos se pueden distinguir: los nodos medioambientales que miden temperatura y humedad del aire, los nodos de agricultura que adicionalmente a las medidas anteriores miden también la humedad y temperatura del suelo; y finalmente, las estaciones medioambientales, midiendo los cuatro valores anteriores, así como presión atmosférica, radiación solar, precipitación caída y velocidad y dirección del viento.

Figura 3.11. Detalle de un cluster de riego en el parque de Las Llamas

Los dos detalles de diferentes *clusters* que se han presentado en la figura 3.10 y figura 3.11, forman parte del despliegue completo mostrado en la figura 3.2.

3.2.4.2 Nodos móviles

La monitorización medioambiental móvil presenta una arquitectura hardware diferente a la mostrada en el apartado anterior, principalmente asociada con la movilidad de los nodos desplegados. A continuación, se muestran los módulos que componen dicha arquitectura:

Figura 3.12. Placa de sensores, módulo CAN-Bus y waspmote

- Placa de sensores: Desarrollado por la empresa española DENIPA [DENIPA], es responsable de la medida de los parámetros medioambientales correspondientes (temperatura, humedad, CO, PM10, O3, NO2), que se envían a la unidad local de proceso. Tanto la alimentación de esta placa, como la transmisión/recepción de datos a/de la unidad local de proceso se realiza mediante un conector RJ45 específico (comportándose como un puerto serie). Además, la placa incluye un controlador básico de tipo RISC (*Reduced Description Set Computer*) a 8MHz, para ejecutar ciertas operaciones sencillas.
- Módulo de bus CAN (*Controller Area Network*): Módulo desarrollado por la empresa española FAGOR Electrónica [FAGOR]. Este módulo se encarga de tomar del bus CAN [CAN Std.] del vehículo los principales parámetros (posición GPS, altitud, velocidad, rumbo y cuentakilómetros)

asociados al mismo, enviándolos a la unidad local de proceso. Es importante resaltar que la toma de datos del bus CAN, se realiza sin violar la garantía del vehículo, es decir sin modificar a nivel *hardware* el bus de datos. Para ello, se utiliza un dispositivo en forma de pinza, que permite tomar (de forma inductiva) los datos que se están transmitiendo a través del bus CAN. El módulo de bus CAN se conecta directamente a la unidad local de proceso.

- Placa *waspmote*: Como se comentó anteriormente, esta placa provee capacidad limitada de procesamiento y de memoria. Para este caso de uso, a diferencia de lo descrito anteriormente en los repetidores estáticos, en este caso la placa de sensores no se conecta a este dispositivo, sino directamente a la unidad local de proceso. Por otro lado, en lugar de las dos interfaces radio indicadas anteriormente, la placa *waspmote* sólo está provista de la interfaz 802.15.4 nativa, para dar soporte a la experimentación (a nivel de nodo), cuando los dispositivos móviles se encuentren dentro de la zona de cobertura del despliegue estático. Los datos referentes a los correspondientes experimentos, se envían a través del puerto serie a la unidad local de proceso.
- Unidad local de proceso (ULP): Este módulo, denominado *CLV*, también desarrollado por FAGOR, se encarga de gestionar tanto la provisión de servicio (datos devueltos por la placa de sensores y el módulo CAN-BUS), la experimentación (envío de los mensajes de *log* asociados a los experimentos realizados sobre la interfaz 802.15.4) y la gestión de la red (reprogramaciones remotas, transmisión/recepción de comandos de configuración). La ULP presenta un procesador RISC de 32 bits a 70 MHz ejecutando un sistema operativo Linux con una memoria Flash de 8MB para aplicaciones de usuario y una RAM de 16MB. Respecto a las interfaces de comunicación, la ULP provee interfaces CAN y RS232/485, 7 entradas digitales y 2 analógicas, así como 5 salidas digitales. Además, este módulo también provee módulo GPS para la georeferenciación de las medidas e interfaz GPRS para ofrecer una cobertura de comunicación global.

Mientras que en los autobuses se han instalado la placa de sensores, el módulo CAN-BUS, la placa *waspmote* y la ULP, en los taxis y los vehículos de parques y jardines, principalmente por motivos de espacio, sólo se instalan las placa de sensores y la ULP, por lo que en estos dispositivos la experimentación a nivel de nodo no está soportada.

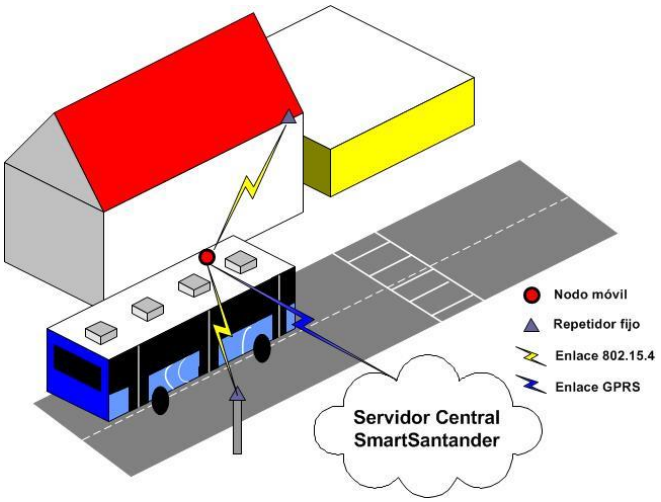

Figura 3.13. Arquitectura hardware del despliegue de nodos móviles

En la figura 3.13, se observa un nodo móvil instalado en un autobús (los únicos que permiten experimentación a nivel de nodo), que se encuentra equipado con una interfaz GPRS que envía

directamente la información a la plataforma central de SmartSantander, mientras que la interfaz 802.15.4 nativa puede interactuar con las correspondientes interfaces 802.15.4 instaladas en los repetidores ubicados en farolas y fachadas. Lógicamente, esta comunicación a través de la interfaz 802.15.4 nativa también se podría llevar a cabo con otro nodo móvil, siempre que estuviera dentro del área de cobertura. En este caso, para asegurar la correcta transmisión de los resultados del experimento, éstos se envían mediante la interfaz GPRS.

La figura 3.14 muestra un detalle de la instalación de los diferentes módulos dentro de un autobús, donde parte de la electrónica se ubica en el interior del vehículo, mientras que la placa de sensores se instala en el exterior (sobre el techo), para poder llevar a cabo la medida de los correspondientes parámetros medioambientales.

Figura 3.14. Instalación en autobuses en el interior (izquierda) y exterior (derecha)

Figura 3.15. Despliegue de nodos móviles en la ciudad de Santander

En la figura 3.15, se muestra un detalle del despliegue completo de nodos móviles, indicando los diferentes tipos de vehículos (autobuses, taxis y vehículos de parques y jardines), en los que se ha realizado la instalación. Como detalle, se muestran los valores medidos y devueltos por uno de los autobuses.

3.2.5 Características principales de la plataforma *software*

Una vez analizados los diferentes casos de uso desplegados con la correspondiente arquitectura hardware, así como las diferentes aplicaciones asociadas a los mismos desarrolladas dentro del proyecto, se presenta la plataforma software sobre la que se implementan dichos servicios/aplicaciones [Sánchez13], [Galache13b]. La arquitectura presentada dentro de SmartSantander, sigue el esquema típico de una red jerárquica con tres niveles: nodo IoT, pasarela y servidor.

El nivel de nodo IoT engloba la mayoría de los dispositivos desplegados en la infraestructura de red. Se compone de diversos dispositivos heterogéneos, incluyendo una amplia pléyade de plataformas de sensores, dispositivos a medida para servicios específicos así como etiquetas RFID y NFC. Estos dispositivos se encuentran típicamente limitados en términos de tamaño de memoria, capacidad de procesamiento y alimentación (baterías de duración limitada), ofreciendo la posibilidad de albergar un gran rango de sensores así como, en ciertos casos, de actuadores. Además estos nodos pueden comportarse o bien como nodos finales, capaces de generar y recibir información, pero sin ofrecer la posibilidad de reenviar los paquetes recibidos de otros nodos; o bien como repetidores, ofreciendo esta capacidad de reenvío necesaria para la comunicación a múltiples saltos asociada a topologías de red mallada.

El nivel de pasarela enlaza los dispositivos IoT con la infraestructura de red principal, agrupándose los nodos IoT en grupos (denominados *clusters*) que dependen de una determinada pasarela a la que se encuentran asociados. El nodo pasarela, por un lado almacena y procesa la información devuelta por los nodos IoT pertenecientes a su *cluster*, y por otro también los gestiona (envío/recepción de comandos), contribuyendo a una gestión más sencilla y escalada de la red completa (compuesta por varios *clusters*). Los nodos pasarela son habitualmente más potentes que los nodos IoT en términos de tamaño de memoria y de capacidad de procesamiento, así como ofreciendo también interfaces de comunicación más potentes y robustas. Además, las pasarelas también permiten la virtualización de dispositivos IoT, habilitando la instanciación de sensores o actuadores emulados que se comportan como los dispositivos reales.

Es importante reseñar que ciertos dispositivos como los teléfonos móviles, así como aquellos de propósito específico con capacidades de computación razonables, se comportan como nodos IoT en términos de capacidades de medida, pero como pasarelas en lo que se refiere a su capacidad de procesado, memoria y comunicación.

El nivel de servidor provee capacidades de cómputo adicionales, ofreciendo una mayor disponibilidad y fiabilidad, encontrándose directamente conectados a la red troncal. Los servidores se utilizan para albergar los repositorios de datos IoT, así como las aplicaciones y servicios correspondientes, recibiendo la información pertinente de las diferentes pasarelas. Adicionalmente, el concepto de federación también es soportado dentro de la arquitectura, donde servidores gestionando redes ubicadas en diferentes localizaciones físicas, pueden conectarse entre ellos para permitir a los usuarios de la plataforma acceder de manera transparente a todos los nodos IoT disponibles, independientemente de la ubicación física del despliegue en el que se encuentre cada nodo.

La arquitectura de SmartSantander se basa en algunos componentes ya existentes, complementados por ciertos bloques desarrollados dentro del proyecto. Las principales plataformas sobre las que se construye la facilidad experimental desarrollada provienen de los proyectos europeos FP7 SENSEI y WISEBED y de la plataforma Telco2.0 [Telco] provista por Telefónica. En la figura 3.16 se muestra la arquitectura de alto nivel con las principales funcionalidades provistas y asociadas con cada uno de los niveles previamente indicados.

La arquitectura mostrada en la figura 3.16 distingue tres subsistemas o planos asociados a la gestión, la experimentación y las aplicaciones, así como uno transversal relacionado con la autenticación y la autorización de los usuarios. Para acceder e interactuar con estos subsistemas, se definen cuatro interfaces, denominados interfaz de soporte a la gestión (ISG), interfaz de soporte a las aplicaciones (ISA), interfaz de soporte a a experimentación (ISE) e interfaz de control de acceso (ICA), respectivamente. Para cada uno de los tres niveles de la arquitectura, se implementan las correspondientes funcionalidades y servicios asociados a cada uno de los cuatro subsistemas.

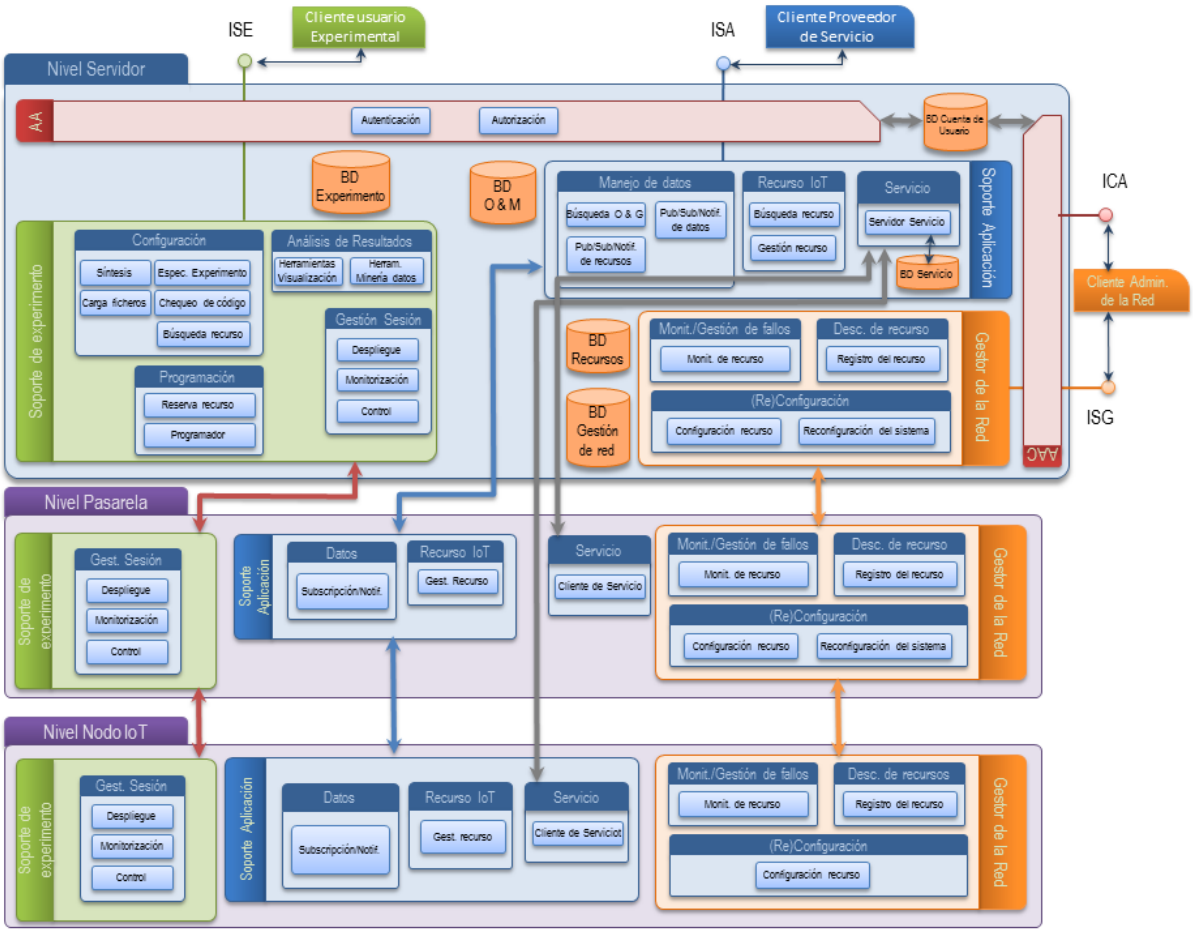

Figura 3.16. Arquitectura a alto nivel y bloques constituyentes de la plataforma de SmartSantander

El subsistema de gestión de la red (SGR) desempeña principalmente, tres procesos de gestión: descubrimiento de recursos, monitorización de recursos y reconfiguración de la red. El descubrimiento de recursos constituye una característica esencial de una plataforma IoT proveyendo soporte para la selección de los recursos en función de los criterios del usuario, tales como los tipos de fenómeno medidos, la ubicación de los sensores a utilizar, la frecuencia de envío de las medidas. Esto esencialmente implica i) la definición de la diversidad de los recursos IoT utilizando un modelo

para la descripción de los recursos, así como ii) la generación de estas descripciones basadas en los registros dinámicos de los nodos IoT, y iii) su búsqueda basada en los correspondientes atributos y las características de conectividad de los dispositivos IoT. Incluso bajo condiciones normales de funcionamiento, la red se encuentra en un estado constante de cambio, con nuevos nodos que se unen a la red, así como otros que la abandonan, presentando un entorno dinámico donde se ha de garantizar la resistencia a fallos mediante la monitorización continúa del estado de los recursos IoT.

Las funcionalidades principales del SGR a nivel de IoT y de pasarela, se asocian principalmente al descubrimiento de los recursos y la monitorización/gestión de los fallos de la red, con la consiguiente reconfiguración de la misma. A nivel de servidor, se encuentran las bases de datos de recursos y de gestión de la red, la primera de ellas almacena información relativa a los dispositivos desplegados (posición, atributos de medida), actualizándose con los nuevos recursos añadidos y aquellos que han experimentado cambios; mientras que en la segunda se almacenan aquellos recursos de la red que se pueden gestionar mediante el envío/recepción de comandos o la reprogramación remota.

El subsistema de soporte a la experimentación (SSE) provee el mecanismo requerido para soportar las diferentes fases del ciclo de vida de experimentación. Durante la fase de especificación, asociada principalmente con la selección de los recursos (p.e. dispositivos IoT) más adecuados para la ejecución del experimento deseado, ofrece al usuario aquellos recursos que se encuentran disponibles (no reservados por otro experimento) durante la duración del correspondiente experimento, seleccionando y reservando aquellos nodos que cumplan y ofrezcan las capacidades demandadas por el experimento a realizar. Una vez que los nodos seleccionados para un determinado experimento se han reservado, comienza el proceso de reprogramación inalámbrica de los nodos seleccionados con la correspondiente imagen de código (asociada al experimente que se desea ejecutar). Finalmente, durante la fase de ejecución, el experimentador dispone de las correspondientes herramientas y comandos para el control de la ejecución y la monitorización del experimento, así como el almacenamiento y el registro de los datos devueltos. Finalmente, a nivel del servidor, la base de datos de experimento se utiliza para almacenar los datos correspondientes a cada uno de los experimentos, ofreciéndolos al experimentador correspondiente.

El subsistema de soporte a las aplicaciones (SSA) se encarga de la provisión de las correspondientes funcionalidades para facilitar el desarrollo de servicios basados en la información almacenada, proveniente de los nodos IoT desplegados. Adicionalmente al almacenamiento de las observaciones y medidas de los nodos IoT en la base de datos O&M (observaciones y medidas), sus funcionalidades principales se refieren a la búsqueda y provisión de estas observaciones a los servicios solicitantes, mediante interacciones de tipo publicación, subscripción y notificación. Por otro lado, para cada uno de los servicios/casos de uso desplegados, se implementan los correspondientes servidores y bases de datos asociados a cada uno de estos servicios.

Finalmente, el subsistema de autenticación y autorización (SAA) ofrece las funcionalidades de control de acceso y autenticación, que se llevan a cabo de manera transversal para proteger todos los puntos de interacción que ofrece la plataforma al mundo exterior. Esta funcionalidad se desempeña a nivel de servidor para controlar y restringir el acceso a experimentadores, proveedores de servicio o administradores de red, autenticados y autorizados contra la plataforma. El almacenamiento de las correspondientes credenciales de acceso, así como de los distintos perfiles de usuario, se realiza en la base de datos de las cuentas de usuario.

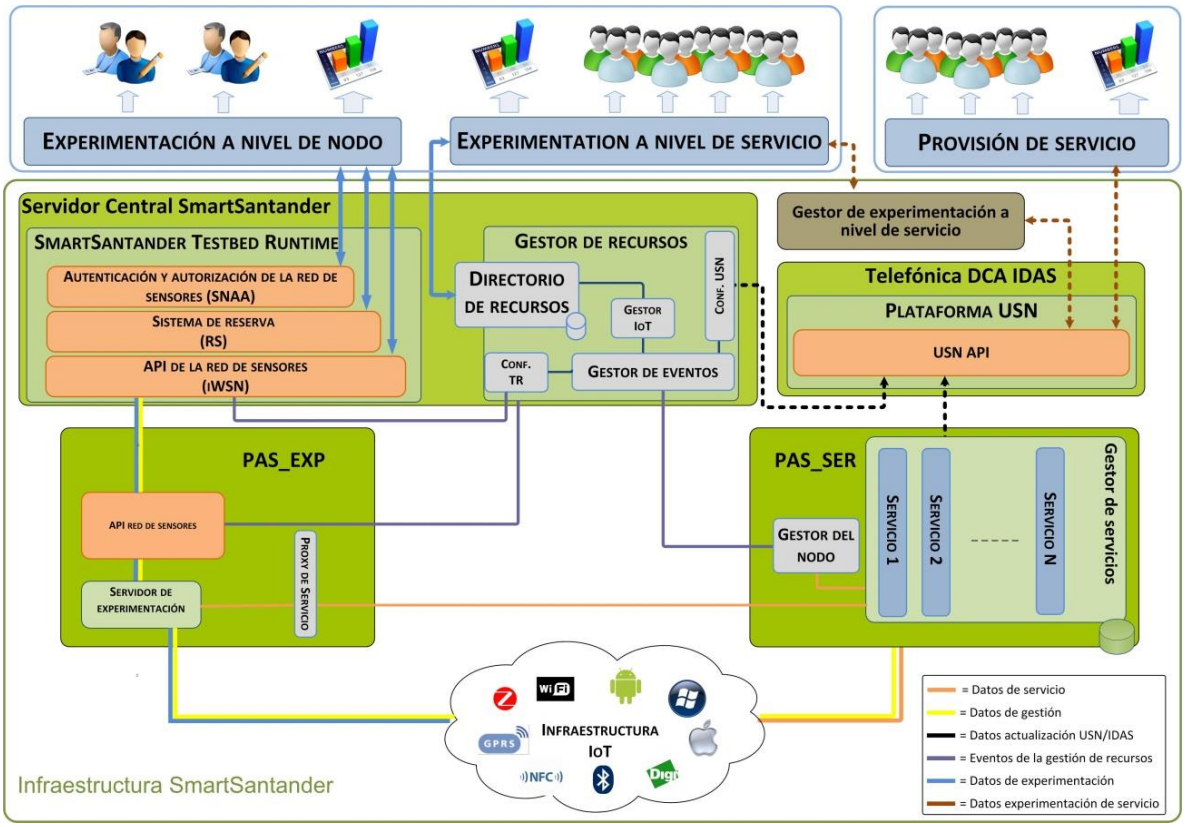

Figura 3.17. Arquitectura a bajo nivel y bloques constituyentes de la plataforma de SmartSantander

La figura 3.17 muestra la arquitectura a bajo nivel que mapea las funcionalidades y servicios previamente descritos sobre bloques constitutivos específicos y está constituida por 5 componentes principales:

- *Servidor central SmartSantander*: Se encarga del manejo de todas las comunicaciones relativas a la experimentación a nivel de nodo llevada a cabo sobre la plataforma, así como de la gestión de los diferentes dispositivos instalados. Para ello, se distinguen dos módulos principales:
  - *SmartSantander Testbed Runtime (TR)*: Este módulo representa el punto de acceso a la plataforma SmartSantander para los administradores del sistema y los experimentadores. Aloja una versión adaptada del *Testbed Runtime* provisto por WISEBED, incluyendo los módulos de autorización y autenticación de la red de sensores (*Sensor Network Authentication and Authorization, SNAA*), el sistema de reservas (*Reservation System, RS*) y la API para la gestión de la red de sensores (iWSN). El SNAA es un componente que ofrece las funciones básicas de control de acceso mediante una autenticación y autorización basada en *Shibboleth* [Shibboleth]. La funcionalidad principal del Sistema de Reservas es la de realizar y gestionar la reserva de los distintos nodos desplegados en la red para llevar a cabo un determinado experimento sobre ellos, soportando la persistencia de estas reservas durante el tiempo requerido por el experimento, así como el mantenimiento de las mismas en las correspondientes bases de datos (BBDD). La API iWSN implementa el conjunto de funcionalidades requeridas para la interacción con los nodos IoT mediante el uso de diferentes comandos, tales como el reseteo, la reprogramación, el escaneo de nodos disponibles o la adición/borrado de enlaces virtuales. El API iWSN también provee la

implementación de un canal para intercambiar mensajes de depuración y control entre el servidor central y la pasarela de experimentación (PAS_EXP), la cual se erige en punto de acceso hacia los correspondientes nodos IoT.

- *Gestor de recursos*: Este módulo permite mantener almacenados los atributos y características propias de cada uno de los nodos desplegados, así como mantener actualizados el estado de los mismos. Para ello, se definen cinco componentes distintos que interactúan para proveer la funcionalidad descrita. A continuación se detallan todos ellos, así como la interacción entre los mismos:
  - *Directorio de recursos*: Heredado del proyecto SENSEI [SENSEI], se erige como un directorio en el que aparecen las propiedades y atributos de los recursos ofrecidos por la plataforma, de forma que se comporta como nexo entre los recursos con sus correspondientes atributos, propiedades, parámetros de medida ofrecidos, y los clientes potenciales que buscan unas funcionalidades particulares.
  - *Gestor IoT*: Se encarga de gestionar las altas de un nuevo nodo o pasarela que se conecte a la red, así como el mantenimiento de los mismos en ésta, mediante el manejo de temporizadores para delimitar el tiempo de validez de un determinado nodo/pasarela. Tanto para el alta de un dispositivo, como para su mantenimiento, el gestor IoT se alimenta de la información enviada desde el gestor de los nodos ubicado en la pasarela para servicio (PAS_SER). El gestor IoT actualiza el directorio de recursos en función del alta/baja de nodos/pasarelas que se haya producido en la red.
  - *Configurador del TR*: Módulo que se encarga de actualizar los ficheros de configuración del TR del servidor central y de la pasarela para experimentación con la información proveniente del gestor IoT, manteniendo actualizado el estado de los nodos que se encuentran disponibles para poder realizar experimentación sobre ellos. Este módulo realimenta, tanto a la instancia del iWSN, incluida en el SmartSantander TR como a aquella ubicada en cada uno de las PAS_EXP específicas para cada servicio (que ofrezca la capacidad de experimentación a nivel de nodo).
  - *Configurador del USN (Ubiquitous Sensor Network)*: Componente que se encarga de actualizar el módulo USN, mediante el alta de los nuevos nodos de servicio que se desplieguen, y la baja de aquellos que no funcionen correctamente y no sigan enviando tramas de servicio al USN. Más adelante se explica en profundidad la funcionalidad del USN.
  - *Gestor de Eventos*: Módulo que se encarga de gestionar todos los eventos, previamente detallados, relacionados con la gestión de recursos; principalmente los asociados con la alta/baja de nodos y pasarelas en la plataforma. Una vez gestionados estos eventos, este módulo interacciona con todos aquellos módulos involucrados en la gestión de recursos.

- Pasarela para experimentación (PAS_EXP): Este tipo de pasarela sólo es necesaria en aquellos casos de uso que permiten la experimentación a nivel de nodo. La comunicación con este módulo es llevada a cabo siguiendo el esquema de formato de mensajes definido por los denominados *protocol buffers* [ProtoBuf]. Los *protocol buffers* constituyen una solución flexible, así como un mecanismo eficiente y automatizado para serializar datos estructurados - similar a XML, pero más pequeño, más rápido y sencillo. El usuario define la estructura de datos que quiere utilizar y mediante un código fuente especialmente generado se puede, de manera sencilla, escribir y leer los datos con la correspondiente estructura a/desde una gran variedad de flujos de datos, así como de lenguajes de programación.

La parte correspondiente de la API iWSN alojada en la PAS_EXP permite la comunicación con el API iWSN que se encuentra en el servidor central, enviando la información asociada a cada uno de los nodos gestionados por cada una de las PAS_EXP. Dentro de cada uno de los nodos desplegados en la red, se instala una instancia del iWSN más ligera que la implementada en la PAS_EXP y en el servidor central, adaptada a las limitaciones en términos de procesado y memoria de los nodos desplegados. Esta instancia permite la gestión adecuada de los comandos enviados a cada uno de estos nodos, tales como los mensajes asociados a la reprogramación inalámbrica.

Por su parte, el módulo de comunicaciones se encarga de tomar los datos provenientes de los sensores desplegados, enviándolos a las aplicaciones/experimentos que los requieran, mediante la multiplexación de varios puertos virtuales sobre un puerto serie físico.

Como se ha comentado anteriormente, la pasarela de experimentación sólo se implementa en aquellos casos de uso que permiten la experimentación a nivel de nodo. En este sentido, la pasarela recibe, tanto los datos de experimentación como los de gestión y servicio. Los dos primeros se envían a través del API iWSN hacia el servidor central, para su correspondiente procesado/reenvío según corresponda. Respecto a los datos de servicio, éstos se envían desde el módulo de comunicaciones al intermediador (*proxy*) de servicio, que los procesa y los reenvía directamente al gestor de servicio de la pasarela para servicio. Es necesario resaltar que para aquellos nodos que no permiten la experimentación a nivel de nodo, los datos de servicio se envían directamente al Gestor de Servicio.

- Pasarela para servicio (PAS_SER): La finalidad de este módulo es la gestión de la información proveniente de los casos de uso, enviando el gestor de servicio estos datos, a las correspondientes instancias de cada uno de los servicios, así como al gestor de nodos. Cada una de las instancias que gestiona cada servicio, procesa los datos recibidos enviándolos al USN. El gestor de nodos utiliza la información recibida (enviada habitualmente de manera periódica) para, a través del gestor de eventos, actualizar el gestor IoT. Éste, mediante los correspondientes temporizadores que tiene habilitados, puede realizar el mantenimiento de todos los nodos, dándolos de alta o baja en el directorio de recursos, así como actualizando los configuradores del TR y del USN.
- Plataforma USN: Esta plataforma es una de las partes de la plataforma global DCA (*Data Collection and Analisys*) IDAS (*Intelligence Data Advanced Solution*), desarrollada por Telefónica para la gestión eficiente de las ciudades inteligentes. La plataforma DCA IDAS [DCA_IDAS] se erige como el principal recurso del facilitador genérico de FI-WARE denominado '*IoT Back-end Device management*' [GE_DCA]. La plataforma USN se caracteriza por ser una plataforma abierta extremo a extremo destinada a ser utilizada en una amplia gama de escenarios y servicios de aplicación, basados en la Internet de las cosas. Esta plataforma presenta una API de tipo REST (*Representational State Transfer*) que ofrece diferentes servicios web implementados en los siguientes bloques funcionales:
  - Suscripción: Permite la suscripción de una determinada aplicación/cliente a determinados eventos, ya sean de tipo 'observación' para la recepción de los valores devueltos por un sensor o conjunto de sensores, o bien de tipo 'registro' asociado a la devolución de un valor cuando se cumple una determinada condición.
  - Notificación: Este bloque se encarga de notificar a la aplicación/cliente, la correcta activación de los correspondientes eventos suscritos, tanto de tipo 'observación' como de tipo 'registro'.

  - Comando: Este módulo permite el envío, por parte del usuario, de diferentes comandos hacia un determinado dispositivo o conjunto de dispositivos, siendo enviada la respuesta de los mismos hacia el usuario correspondiente.
  - Datos de los sensores: Este bloque facilita la petición y provisión de los datos almacenados en el USN. En este sentido, se habilitan diferentes métodos de tipo obtener (*get)*, añadir *(add)*, actualizar *(update)*, borrar *(delete)*, que permiten actuar sobre los datos almacenados.
  - Cliente: Este módulo permite añadir, actualizar y eliminar datos propios de un determinado usuario.
- Gestor de Experimentación a nivel de servicio: Módulo intermedio entre la plataforma USN y el experimentador a nivel de servicio, que permite a este experimentador reservar el conjunto de nodos correspondiente a su experimento. De esta forma, el experimentador tiene acceso a los datos de los nodos correspondientes a su reserva a través de las diferentes interfaces ofrecidas por la plataforma USN.

En la figura 3.17, se identifican tres tipos de usuarios externos de la plataforma: los proveedores de servicio, los experimentadores a nivel de nodo y aquellos que realizan experimentación a nivel de servicio.

- Como parte de la provisión de servicio se incluyen los casos de uso desarrollados dentro del proyecto SmartSantander. Cada uno de estos casos de uso conlleva el despliegue de un determinado número de dispositivos IoT con diferentes capacidades de medida. Los datos devueltos por la infraestructura IoT, son recogidos por la pasarela de servicio, en cada uno de los módulos asociados a cada servicio que procesan esta información y la envían a la plataforma USN. La información asociada a todos estos servicios será ofrecida posteriormente a los experimentadores a nivel de servicio.
- Los experimentadores nativos acceden a la plataforma, a través del módulo SNAA, para identificarse contra la plataforma y poder acceder a los recursos que permiten la experimentación a nivel de nodo (no todos los dispositivos desplegados ofrecen esta capacidad). Después de autenticarse, el usuario accede al sistema de reservas para poder reservar el conjunto de dispositivos que necesite, y la duración determinada de la reserva en función del experimento a llevar a cabo. Finalmente, el usuario carga el código correspondiente al experimento de manera inalámbrica (a través del OTAP) en los nodos reservados. La interacción entre el servidor central, la PAS_EXP y los nodos reservados, es llevada a cabo a través de las instancias del iWSN instaladas en cada uno de estos dispositivos. Por último, los datos generados por el experimento, son enviados desde el módulo iWSN del servidor central al experimentador correspondiente.
- Los experimentadores a nivel de servicio se autentican contra la plataforma USN, comunicándose con ella a través de los diferentes servicios Web que ofrece el USN, para poder suscribirse a los correspondientes eventos de tipo ‘observación’ y ‘registro’, asociados al nodo o conjunto de nodos involucrados en el servicio/experimento a realizar. Además, teniendo en cuenta las funcionalidades ofrecidas por el USN también se pueden ejecutar diferentes comandos sobre los nodos desplegados.

Por último, dentro del marco del proyecto, también se han desarrollado ciertas aplicaciones/servicios (sensado participativo y realidad aumentada), que al igual que ocurre con la experimentación a nivel de servicio, acceden a los datos de servicio provistos por el USN. De manera particular, el sensado

participativo, presenta una doble vertiente puesto que no sólo se alimenta de los datos devueltos por el sistema, sino que también alimenta al mismo inyectando, de manera anónima por parte de los usuarios, los datos de los sensores con los que los nuevos teléfonos inteligentes se encuentran provistos. Es el denominado concepto de *prosumer* (del inglés *producer* + *consumer*), en el que los usuarios producen y consumen información para/de la plataforma.

Una vez definida la arquitectura global de la plataforma, se deben definir ciertas particularidades asociadas a los casos de uso en los que se basa este trabajo (ofrecen la posibilidad de experimentación a nivel de nodo). En este sentido, al igual que en la parte hardware, a nivel *software* se ha de diferenciar entre los nodos estáticos y los móviles, indicando las características y peculiaridades asociadas a cada uno de ellos.

3.2.5.1 Nodos estáticos

Teniendo en cuenta la arquitectura *software* global de la plataforma y el despliegue hardware anteriormente descritos para este caso de uso, la arquitectura asociada al mismo, se ilustra en la figura siguiente:

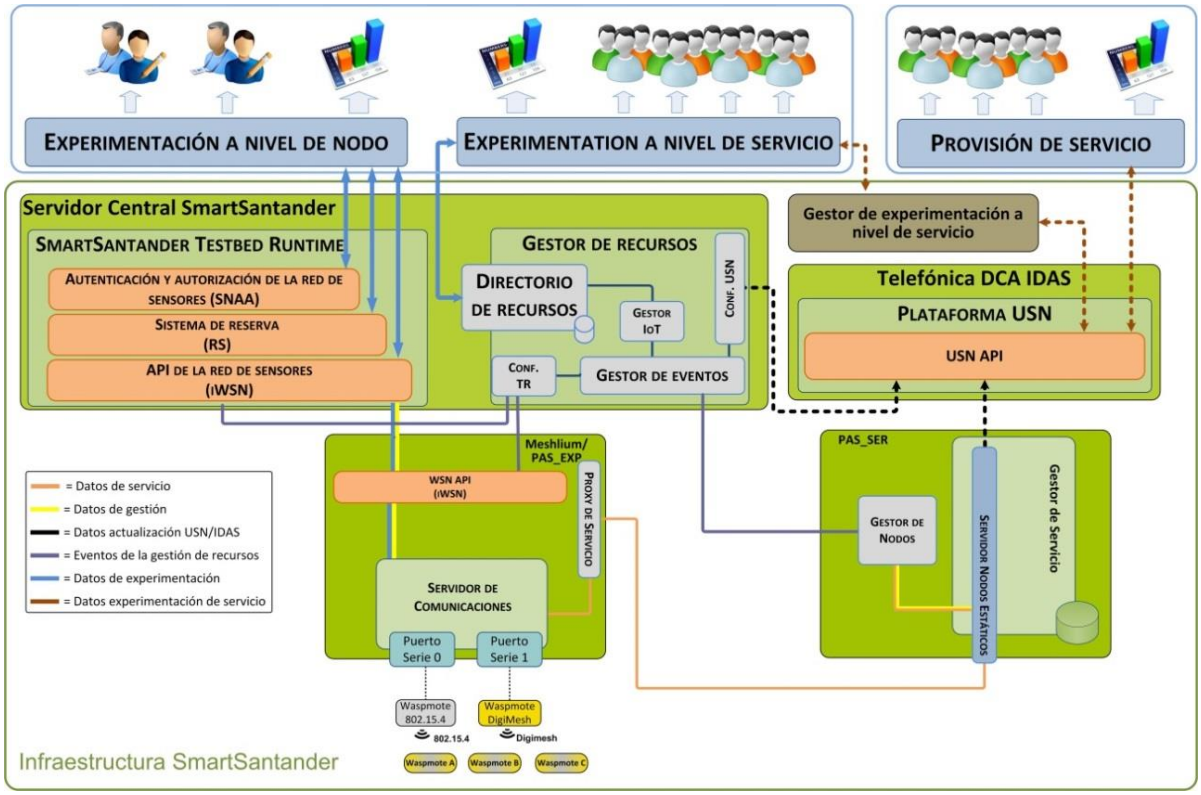

Figura 3.18. Arquitectura software para los casos de uso de monitorización medioambiental estática y riego inteligente

En la figura 3.18 se pueden destacar los siguientes aspectos:

- Se mantienen los usuarios a nivel de nodo y a nivel de servicio, puesto que este caso de uso permite llevar a cabo estos dos tipos de experimentación.
- Cada una de las pasarelas desplegadas se corresponde una pasarela de experimentación que se encarga de gestionar el conjunto de nodos que se encuentran dentro de su *cluster*. En este

sentido, cada pasarela posee una instancia de la API iWSN, para interactuar con el servidor central y manejar adecuadamente los mensajes asociados, tanto a la gestión de la red como a la experimentación a nivel de nodo. En lo que se refiere a la información relacionada con la provisión de servicio, el *proxy* de servicio se encarga de enviarla a la correspondiente instancia del gestor de servicio dentro de la pasarela para servicio.

- Respecto al módulo de comunicaciones, éste es denominado servidor de comunicaciones, el cual consiste en un multiplexor de puertos que ofrece varios puertos de comunicación virtuales sobre una interfaz física única.

Analizando la figura 3.19 de 'abajo-arriba', se observan dos interfaces físicas (puertos serie) disponibles en cada pasarela, una de ellas (puerto serie 0) asociada a la transmisión/recepción de datos de experimentación, mientras que la otra (puerto serie 1) se encarga del manejo de los datos asociados a la provisión de servicio y gestión de la red. El servidor de comunicaciones ofrece diferentes puertos virtuales multiplexados sobre los puertos serie físicos. En este caso, según se muestra en la figura, aparecen diferentes puertos asociados a la experimentación (EXP 1:N), varios al servicio (SER 1:N) y uno a la gestión de la red (asociado al OTAP). Los casos de uso de monitorización ambiental estática y riego inteligente utilizan un mismo puerto virtual de servicio, el cuál recibe información periódica (cada 5 minutos), de los diferentes parámetros medidos por los dispositivos desplegados.

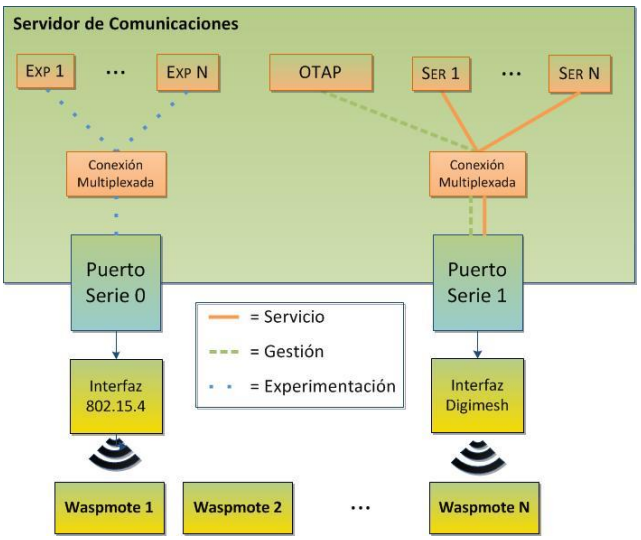

Figura 3.19. Servidor de comunicaciones

Cabe recordar nuevamente que, aunque existe una interfaz 802.15.4 nativa en las pasarelas para gestionar las comunicaciones 802.15.4, el tráfico asociado a los resultados obtenidos de la experimentación se envía a través de la interfaz *Digimesh*. Esto es debido a que la interfaz 802.15.4 nativa no provee un protocolo de enrutamiento nativo y, por lo tanto los nodos fuera de la zona de cobertura (a más de un salto) de la pasarela, no son capaces de comunicarse con él. Esto se traduce en que se utiliza la interfaz 802.15.4 para llevar a cabo el experimento, pero para asegurar el envío de los resultados obtenido se utiliza la interfaz *Digimesh*. Lógicamente, esto implica que se han de generar los correspondientes puertos virtuales (asociados al correspondiente puerto físico) a nivel de servicio/gestión, para enviar al experimentador los datos correspondientes a su experimento. No obstante, la transmisión y manejo de los nodos asociados a un experimento se explicará de manera detallada en el Capítulo 5.

- Los datos de experimentación y gestión de la red son enviados hacia el TR ubicado en el servidor central, mientras que los datos asociados al servicio se envían (a través del *proxy* de servicio) hacia el módulo correspondiente (servidor de nodos estático), ubicado en el gestor de servicio.
- El servidor para nodos estáticos se encarga de enviar los datos de servicio (recibidos a través del Proxy de Servicio) a la plataforma USN. Por otro lado, esta información recibida de forma periódica (cada 5 minutos), es utilizada por el gestor de nodos para el mantenimiento de los nodos de la red, enviando al gestor IoT el evento correspondiente cuando recibe una trama de servicio de un nodo desplegado. Con la recepción del evento específico, el gestor IoT actualiza el temporizador correspondiente al nodo de servicio que envío la trama. Si es la primera vez que un nodo envía una trama de servicio, el gestor IoT lo dará de alta en el directorio de recursos, actualizando los ficheros de configuración del USN (a través del configurador del USN) y del TR de la pasarela correspondiente (a través del configurador de TR). Por último, además de la actualización del estado de los nodos, el gestor de nodos también envía un mensaje cada cierto tiempo referente al estado de cada una de las pasarelas, de forma que también sean actualizados adecuadamente a través del gestor IoT.

Una vez indicada la forma en que, desde el punto de vista de la pasarela y del servidor central, se ofrece la provisión de los determinados servicios, y se habilita la capacidad de experimentación con la red, así como se lleva a cabo la gestión de la misma, se detalle en la siguiente figura se detalla la arquitectura, a nivel de nodo, para desempeñar las mencionadas funcionalidades.

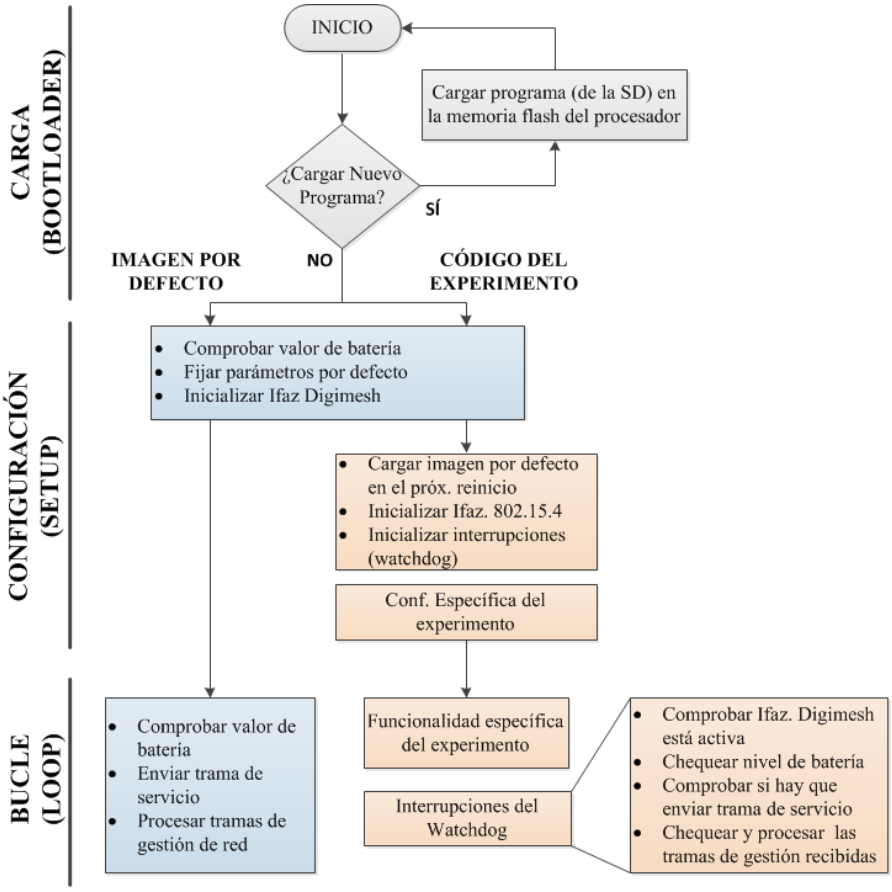

Figura 3.20. Arquitectura software del nodo fijo (waspmote)

Como se puede derivar de la figura 3.20, y teniendo en cuenta que el procesador del que se encuentra provista la placa *waspmote* tiene un único hilo de ejecución, el flujo de funcionamiento del mismo es de tipo secuencial, dividiéndose principalmente en tres etapas: carga (*bootloader*), configuración (*setup*) y bucle (*loop*).

- Carga: El proceso de carga se realiza en el momento de encendido de la placa. Durante este proceso se le indica al procesador el programa que debe cargar cuando se inicie, pudiendo ser el mismo que se estaba ejecutando anteriormente (programa ya cargado en la memoria *flash* del dispositivo), o uno nuevo programa que se encuentra almacenado en la memoria SD externa del dispositivo. En este segundo caso, teniendo en cuenta que dentro de la memoria SD se pueden encontrar almacenados varios programas, el procesador apuntará al inicio del programa correspondiente a ser cargado, volcándolo sobre la memoria *flash* y comenzando su ejecución. Dentro de los programas cargados en el *waspmote*, se pueden distinguir dos tipos: la imagen por defecto (*golden image*) y los programas de experimentación. La imagen por defecto, como ya se ha descrito anteriormente en este capítulo, permite que el nodo se encuentre en un estado confiable y accesible, de forma que se asegura el envío de las tramas de provisión de servicio correspondientes (monitorización ambiental estática o riego, según corresponda), así como la gestión remota de los nodos habilitando los mecanismos necesarios para la recepción, procesado y respuesta de comandos de gestión desde la pasarela. El programa de experimentación, además de las funcionalidades asociadas a la provisión de servicio y gestión de la red, también habilita la interfaz de experimentación (802.15.4 nativa), para poder ejecutar un experimento de manera concurrente.

- Configuración: La parte de configuración se ejecuta una única vez al principio del programa. Como tanto la imagen por defecto, como el código de un experimento, comparten el soporte a la provisión de servicio y a la gestión de la red, ambos inicializan la interfaz *Digimesh* , la cual siempre se encontrará activa para permitir la recepción de comandos de gestión, así como el reenvío de tramas de servicio en cualquier momento, fijando los parámetros por defecto correspondientes, tales como:

    - La frecuencia del canal de comunicación: El protocolo *Digimesh* trabaja a la frecuencia ISM de 2.4 GHz, contemplando el uso de 12 canales diferentes para evitar la colisión entre *clusters* adyacentes. Adicionalmente, para cada uno de los diferentes canales se pueden establecer diferentes identificadores de red (*Personal Area Network Identifier, PANID*), permitiendo la generación de grupos (a nivel físico), dentro de un mismo *cluster*. En este sentido, para cada uno de los *clusters* desplegados dentro de la ciudad se define un canal y un identificador de red, de forma que los *clusters* adyacentes utilizan canales lo más disjuntos posible para evitar interferencias. En la configuración inicial de los nodos se fijan estos parámetros de acuerdo a la pasarela de la que dependan estos nodos.

    - Dirección de la pasarela: Para asegurar que los mensajes de los nodos se envían a la pasarela correspondiente, se almacena la dirección MAC de la misma.

    - Tipo de nodo: En función de las capacidades de medida de cada uno de los nodos, tales como temperatura, luminosidad, ruido, pluviómetro, anemómetro, se almacena el tipo de nodo correspondiente, de forma que la generación de la trama de provisión de servicio se realiza en función del tipo de nodo especificado.

- Intervalo de servicio: Define el intervalo de tiempo entre cada uno de los envío de las tramas de servicio. En principio, este intervalo se encuentra fijado a 5 minutos, pero podría variarse en función de las necesidades del servicio a ofrecer.

Por último, se chequea el valor de la batería de manera periódica para evitar que la batería se descargue completamente (el nodo sólo se podría reactivar mediante un *reset* físico). De esta forma, cuando la batería se encuentra por debajo de un determinado umbral mínimo, se coloca el nodo en un estado de hibernación, de forma que periódicamente éste chequea el estado de la batería (en los ciclos nocturnos se cargará), activando de nuevo el nodo cuando el valor de batería supera un umbral máximo. Para evitar un efecto rebote, los umbrales máximo y mínimo son diferentes, implementándose el correspondiente mecanismo de histéresis. Lógicamente, durante este proceso el nodo no es accesible para ser gestionado, no envía sus tramas de servicio y tampoco se comporta como nodo reenviador de las tramas de otros nodos.

Respecto al código de experimento, además de la configuración previamente descrita, se configura que en el siguiente reinicio remoto del nodo, se cargue la imagen por defecto. Esta es una manera de solventar posibles comportamientos anómalos de un experimento instalado en un nodo, de forma que sea posible reiniciar el nodo con el programa por defecto cuando el comportamiento del experimento no sea el deseado. Desde el punto de vista del funcionamiento de la experimentación, se inicializa la interfaz 802.15.4 nativa definiendo el canal e identificador de red correspondientes. En este caso, a diferencia de la interfaz *Digimesh*, a nivel 802.15.4 todos los nodos de la red se configuran con el mismo canal e identificador de red (detallado en el Capítulo 5). Por otro lado, se habilitan las interrupciones (*watchdog*), que permitirán la ejecución concurrente de la experimentación, de manera conjunta con la provisión de servicio y la gestión de la red. Finalmente, se definirán e inicializarán aquellos parámetros de configuración referentes al experimento específico.

- Bucle: La parte de código que se encuentra dentro de esta sección se ejecuta de manera continúa hasta que el nodo se apaga o se carga un nuevo código. Desde el punto de vista de la imagen por defecto, se realiza la comprobación del nivel de batería (al igual que se señaló en la fase de configuración), y realiza el envío de la correspondiente trama de servicio (en función del tipo de nodo), cuando corresponda en función de la periodicidad de envío asociada a cada servicio. Además, de manera continua, el nodo se encuentra escuchando la posible recepción de tramas de gestión, las cuales serán procesadas, enviando la correspondiente respuesta.

  En lo referente al código de experimento, se define dentro del bucle la funcionalidad correspondiente del experimento, principalmente asociada a la recepción, procesado y envío de las correspondientes tramas de experimentación, a través de la interfaz 802.15.4 nativa. Por otro lado, teniendo en cuenta que el dispositivo presenta un solo hilo de ejecución, cada cierto tiempo (fijado a 1 segundo) se comprueba el nivel de batería, se chequea si se ha cumplido el período de tiempo para el envío de la trama servicio (enviándola si corresponde), así como si se ha recibido una trama de gestión procesándola convenientemente.

Como se puede derivar de la arquitectura de este caso de uso, tanto a nivel de servidor/pasarela como de nodo, ésta permite la realización de experimentación, a nivel de nodo y a nivel de servicio, transmisión/recepción de comandos referentes a la gestión de la red e información asociada a la provisión de servicio, todo ello de manera simultánea a través de la funcionalidad provista por el

servidor de comunicaciones. Por otro lado, es una arquitectura que se realimenta con la información periódica asociada a la provisión de servicio, con la cual se actualiza el directorio de recursos, el iWSN y el TR, con el estado actual de los nodos desplegados en la red.

3.2.5.2 Nodos móviles

El caso de uso de Monitorización medioambiental móvil presenta una arquitectura similar a la planteada para el caso estático, como se muestra en la figura 3.21, aunque existen pequeñas diferencias asociadas principalmente con la movilidad de los nodos y que se definen a continuación:

- La pasarela para experimentación, denominada pasarela para nodos móviles (PAS_MOV), la cual implementa la correspondiente instancia del API iWSN para la gestión de las comunicaciones con los nodos desplegados, mediante el servidor de flotas.

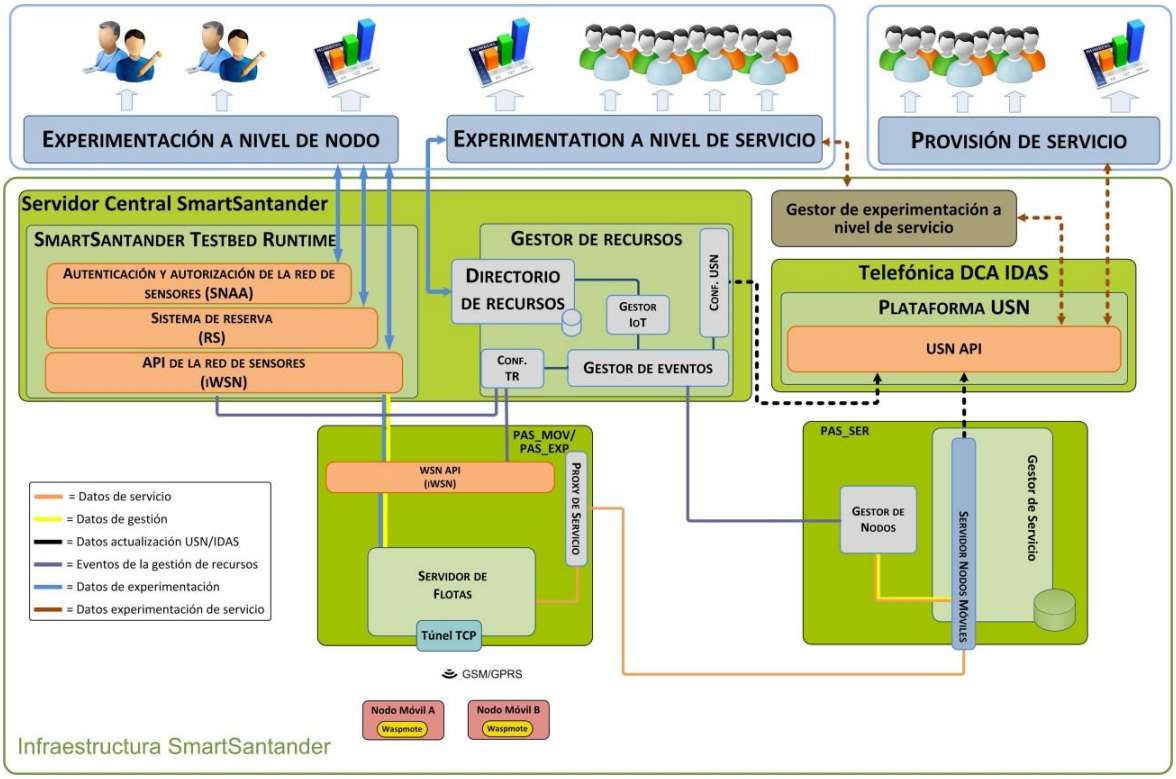

Figura 3.21. Arquitectura software para el caso de uso de monitorización medioambiental móvil

- El servidor de flotas se encarga de recibir, a través de un túnel TCP, los valores devueltos por los nodos móviles a través de la interfaz GSM/GPRS. Habitualmente, los datos son enviados por GPRS, pero en puntos donde no existe cobertura GPRS, se utiliza la interfaz GSM (a través de mensajes SMS) para llevar a cabo la comunicación.
- Al igual que ocurría en el caso estático, los datos de experimentación y gestión de la red son enviados hacia el TR ubicado en el servidor central, mientras que los datos asociados al servicio se envían (a través del *proxy* de servicio) hacia el módulo correspondiente (servidor de nodos móviles), ubicado en el gestor de servicio.
- Los datos de servicio provenientes de la placa de sensores asociada a cada dispositivo, así como la información tomada del bus CAN del vehículo (sólo disponible en algunos autobuses), es enviada de manera periódica. Esta periodicidad de envío, al igual que ocurría en el caso anterior,

permite al gestor de nodos enviar los eventos correspondientes al gestor de recursos ubicado en el servidor central, el cuál mediante la correspondiente interacción con los bloques que lo componen lleva a cabo la gestión de los nodos desplegados (de la misma manera descrita en el caso de los nodos estáticos).

- Respecto a la experimentación a nivel de nodo (disponible sólo en autobuses), al igual que en el caso de uso estático, se habilita la interfaz 802.15.4 para llevar a cabo el experimento, bien interactuando con la red estática desplegada (cuando el vehículo se encuentre dentro de su zona de cobertura) o interactuando con otros módulos 802.15.4 instalados en otros autobuses. En este caso, para enviar los resultados derivados del experimento, se utiliza la interfaz GSM/GPRS para, al igual que en el caso de los nodos estáticos con la interfaz *Digimesh*, asegurar que los resultados del experimento se reciban correctamente en la pasarela para nodos móviles. Esto se traduce en un paralelismo entre las interfaces GPRS y la *Digimesh* en los escenarios móvil y estático, respectivamente.

Una vez indicada la forma en que, desde el punto de vista de la pasarela y del servidor central, se ofrece la provisión de servicio, la capacidad de experimentación y la gestión de la red, se detalla el modo en que se lleva a cabo en el nodo móvil. Este nodo, como se ha descrito anteriormente en este capítulo, se compone de tres elementos principales: unidad local de proceso, placa de sensores y *waspmote*. Es importante reseñar que la comunicación entre los nodos y la pasarela/servidor se realiza siempre a través de la ULP, de forma que la comunicación entre la ULP con el *waspmote* y con la placa de sensores se realiza mediante una configuración maestro-esclavo. Desde el punto de vista de la provisión de servicio, la placa de sensores toma los datos correspondientes de manera periódica, entregándoselos a la ULP bajo petición de la misma (actualmente fijada a un período de 1 minuto). Estos datos, además de enviarse a las correspondientes entidades superiores de la arquitectura (indicado anteriormente), también son enviados por la ULP al *waspmote* para que puedan utilizarse como fuente de datos para llevar a cabo un determinado experimento.

Respecto a la parte de gestión, se ha de distinguir aquella que se realiza sobre la ULP, dirigida a la variación de los períodos de toma de datos de la placa de sensores o la actualización del firmware; con la que se lleva a cabo sobre el *waspmote*, orientada principalmente a la reprogramación del mismo con el/los experimentos que correspondan. La gestión de la ULP se realiza directamente a través de la interfaz GPRS, mientras que para la gestión del *waspmote* la ULP se erige como nodo intermedio (interfaz de comunicación RS-232), reenviando la información recibida desde el servidor central al *waspmote*, y viceversa. De esta forma, el protocolo de reprogramación es similar al utilizado en los nodos estáticos, con la principal diferencia de que en este caso se utiliza una interfaz cableada (más confiable) y que la reprogramación es siempre unicast (cada ULP controla un solo *waspmote*). Finalmente, respecto a la experimentación, el *waspmote* envía los resultados derivados de cada uno de los experimentos al ULP, el cual se encarga de reenviarlos al servidor central y, a su vez, al experimentador correspondiente.

Como se puede observar a nivel arquitectural, el caso de uso móvil es muy similar al estático, excepto por el hecho de que para el caso móvil todos los nodos dependen de la misma pasarela de experimentación (PAS_MOV), mientras en el caso estático, cada *cluster* de nodos está asociado a su correspondiente pasarela de experimentación.

3.3 EL DEVENIR DE LAS CIUDADES INTELIGENTES

Como se ha comentado en los apartados anteriores, la proliferación de despliegues IoT masivos (principalmente en entornos urbanos) está aumentando exponencialmente, estando destinados principalmente a la implementación de servicios orientados a los ciudadanos y a las Administraciones Públicas, y sirviendo como base para el desarrollo de servicios y aplicaciones por parte de los proveedores de servicio y los desarrolladores de aplicaciones; así como siendo el lugar ideal para la experimentación de la comunidad científica en entornos masivos reales. Ésta es la situación actual, pero existen un conjunto de aspectos clave en el devenir de las Ciudades Inteligentes, como se indica a continuación:

- Datos abiertos (open data): Actualmente, todos los datos provistos por las diferentes infraestructuras desplegadas, así como la información generada por los usuarios, se traduce en una amplia cantidad de información, la cual puede hacerse disponible o no a los usuarios. En este sentido, las denominadas políticas de datos abiertos, se caracterizan por:
  - Estandarización: En aras de facilitar el tratamiento, almacenamiento y procesado de los datos devueltos por las diferentes infraestructuras desplegadas, la estandarización del formato de envío de datos, se erige como pilar principal, facilitando el desarrollo de aplicaciones compatibles para todas estas plataformas.
  - Confiabilidad: La información ofrecida a los usuarios debe ser confiable, asegurando que los datos proporcionados se corresponden con valores fidedignos de los correspondientes parámetros medidos.
  - Disponibilidad/Accesibilidad: Los datos obtenidos no puedan ser disponibles y accesibles para todos los usuarios, estableciendo diferentes tipos de usuario, asociados a aspectos técnicos, sociales o económicos.
  - Seguridad: De acuerdo con lo indicado en el apartado anterior, y en función del grado de acceso a la información por parte de un determinado usuario, se ha de garantizar la seguridad de la información proporcionada, evitando accesos fraudulentos a la misma.

Teniendo en cuenta el auge que están experimentando todas las políticas de datos abiertos, los diferentes gobiernos están llevando a cabo iniciativas a nivel internacional, nacional y local, colaborando con diferentes organizaciones de cooperación económica, social y política [OPEN_DATA_SITES].

- Ciudadanos y servicios públicos: La mayoría de los servicios/aplicaciones desarrollados sobre las infraestructuras desplegadas en entornos urbanos se encuentran destinadas a los ciudadanos, bien de manera directa (por ejemplo aplicaciones para teléfonos móviles), o indirecta (a través de las correspondientes Administraciones Públicas). Para este último caso, se hace indispensable la implicación de los gestores tanto a nivel nacional como local, utilizando la información generada para mejorar el funcionamiento actual de los diferentes servicios urbanos. Desde el punto de vista de los ciudadanos, desempeña un papel primordial la concienciación y colaboración de los mismos, conseguida mediante:
  - Encuestas/cuestionarios: A través de los sitios web de las diferentes Administraciones Públicas, se pueden recoger las inquietudes y necesidades de los ciudadanos, así como evaluar el grado de satisfacción de los mismos con los servicios ofrecidos.

  - Talleres prácticos: Estos talleres permiten acercar a los usuarios a los diferentes servicios y aplicaciones que se desarrollan en el ámbito urbano, asociados a las nuevas infraestructuras desplegadas.
  - Generación de eventos/subscripción a notificaciones: Los usuarios no sólo se comportan como consumidores de información, ya sea generada por proveedores de servicios, desarrolladores de aplicaciones o Administraciones Públicas, sino que el usuario también se comporta como productor de información. Surgen los conceptos de medida o urbanismo participativo, donde los usuarios generan eventos con información sobre lo que ocurre a su alrededor, pudiendo estos mismos usuarios suscribirse a notificaciones generadas por otros usuarios.
- Indicadores de rendimiento claves: Conocidos como KPIs (*Key Performance Indicators*), se erigen como los parámetros de referencia para evaluar el grado de servicio asociado a una determinada aplicación/servicio. Teniendo en cuenta este aspecto y para permitir la comparación y la evaluación de diferentes servicios, es necesario generar un conjunto común de KPIs, relacionados con aspectos técnicos, moldeados por los requerimientos de los servicios públicos y realimentados por la opinión de los ciudadanos. Por otro lado, también se deben afrontar los venideros avances técnicos con los correspondientes nuevos KPIs asociados, la generación de reglas y de directrices para el desarrollo de servicios alineados con estos nuevos KPIs, así como la clasificación de proyectos y servicios de acuerdo al grado de cumplimiento de los KPIs asociados a los mismos.
- Mantenimiento y sostenibilidad: Como se ha comentado previamente, la mayoría de las infraestructuras desplegadas se asocian, en general, con el desarrollo de proyectos de investigación y, por lo tanto, con una financiación limitada y acotada en el tiempo. En este sentido, para asegurar la continuidad de la funcionalidad y operatividad de estas plataformas, es necesario el mantenimiento y la sostenibilidad de las mismas después de la finalización del proyecto. Desde el punto de vista del mantenimiento de la plataforma desplegada, los ciclos de vida limitados de la batería de los dispositivos instalados, la actualización del firmware cargado en los mismos, así como el deterioro o fallo a nivel *hardware*, se erigen como los principales aspectos a ser considerados. En lo que se refiere a la sostenibilidad de la plataforma, se han de considerar los gastos de mantenimiento que se extienden más allá de la duración del proyecto, así como el desarrollo de nuevos servicios asociados a capacidades de medida adicionales.

La finalidad de las infraestructuras IoT desplegadas, se dirige a mejorar el rendimiento y la sostenibilidad de los servicios públicos ofrecidos. Actualmente, la mayoría de los servicios urbanos tales como la gestión de residuos urbanos, el transporte público, el alumbrado público o el mantenimiento de parques y jardines, trabajan de manera independiente y aislada. Lógicamente, la mejora de los servicios disponibles actualmente, supone un paso importante, pero éstos siguen adoleciendo de una gestión y un funcionamiento conjunto, de forma que los siguientes pasos se encaminan hacia un enfoque de gestión transversal de los servicios subyacentes.

3.4 CONCLUSIONES

Dentro de este capítulo, se ha presentado un detallado estado del arte indicando las diferentes iniciativas y proyectos que, por parte de la comunidad científica y tanto a nivel nacional como a nivel europeo y mundial, se están desarrollando con el objetivo de profundizar en el estudio de despliegues IoT masivos y la evolución de la Internet del futuro en el entorno de una ciudad inteligente.

De entre todos los proyectos/iniciativas descritos, se ha realizado especial hincapié en el proyecto SmartSantander, el cual incluye un despliegue masivo de 20.000 dispositivos IoT, 12.000 de los cuales se instalan en la ciudad de Santander, sirviendo esta infraestructura como marco principal en el que se encuadra y sobe el que se realiza esta Tesis Doctoral. El proyecto se caracteriza por perseguir un doble objetivo, ofreciendo por un lado un banco de pruebas para la experimentación de nuevas tecnologías y arquitecturas habilitadoras para la IoT, y realizando por el otro, el desarrollo de diferentes servicios y aplicaciones, sobre la arquitectura desplegada, y destinadas a los ciudadanos.

En aras de cubrir la doble vertiente experimentación-servicio mencionada, se ha definido dentro de este capítulo, la arquitectura de alto y bajo nivel con los correspondientes módulos y entidades constitutivos de la misma, que permiten ofrecer la provisión de los diferentes servicios sobre la red desplegada y la capacidad de experimentación sobre la misma, de manera conjunta con la correspondiente gestión remota de los diferentes dispositivos desplegados. Dentro de los diferentes casos de uso desplegados en el marco del proyecto SmartSantander, se presta especial interés a aquellos que permiten realizar experimentación tanto a nivel de servicio como a nivel de nodo, sobre dispositivos fijos y móviles, indicando las correspondientes módulos hardware, las interfaces y protocolos de comunicación, así como las entidades *software* que permiten llevar a cabo las funcionalidades asociadas a cada uno de estos casos de uso seleccionados.

4 GESTIÓN DE RED: REPROGRAMACIÓN VÍA RADIO DE DISPOSITIVOS IOT

Este capítulo aborda la especificación, desarrollo e implementación de un protocolo que permita reprogramar de manera remota un conjunto de dispositivos IoT desplegados en la plataforma SmartSantander.

Primeramente, se realiza una introducción de los diferentes protocolos existentes para reprogramación remota, tomando como base algunos de los mismos para la implementación del protocolo previamente citado. Seguidamente, se detalla su funcionamiento en los diferentes modos de operación (unicast, multicast y broadcast), caracterizándolos a partir de las diferentes medidas de latencia y caudal eficaz para así validar su correcto funcionamiento.

4.1 INTRODUCCIÓN Y ESTADO DEL ARTE

4.1.1 Introducción

Como se ha venido insistiendo durante este trabajo, es una realidad que las infraestructuras de red inalámbricas presentes y futuras se van a encontrar fuertemente influidas por la presencia masiva de dispositivos IoT. Por consiguiente, se antoja necesario ofrecer a la comunidad científica, infraestructuras que permitan analizar y evaluar el rendimiento de mecanismos dirigidos a integrar el mundo IoT en la infraestructura de la Internet del futuro.

Muchos de los proyectos descritos en el Capítulo 3, se caracterizan por la realización de despliegues de redes de nodos de mayor o menor tamaño, ya sea con la finalidad de hacer coexistir tecnologías radios subyacentes heterogéneas, medir diferentes parámetros, o generar servicios para los ciudadanos; en definitiva, para experimentar sobre la red desplegada. En este sentido, se hace necesario dotar a la red de una flexibilidad que la permita responder a los requerimientos específicos que emanan de la reconfiguración de los servicios en función de las necesidades de los ciudadanos, proveedores de servicio, experimentadores, arbitrando los mecanismos necesarios para gestionar el correcto funcionamiento de los nodos. Para proporcionar esta flexibilidad, surge el concepto de la reprogramación inalámbrica, que permite configurar los nodos de manera remota con diferentes códigos, modificando el funcionamiento de aquéllos.

Como se ha explicado, dentro del proyecto SmartSantander y desde el punto de vista de la gestión de la red, se implementan los correspondientes mecanismos para enviar comandos hacia/desde los nodos, así como para reprogramar los mismo tantas veces y con tantos códigos como sea necesario. La capacidad de poder realizar la reprogramación remota de los nodos permite acometer la dualidad experimentación-servicio perseguida por el proyecto, a saber:

- Provisión de servicio: Diferentes configuraciones/tipos de servicio, asociados a las capacidades de medida de un determinado conjunto de nodos, pueden ser definidas mediante la modificación del código instalado en los nodos, respondiendo a los requerimientos asociados a ese determinado servicio/aplicación.
- Experimentación: Distintos experimentos, tales como aquellos asociados a la implementación de diferentes protocolos de enrutamiento, técnicas de minería de datos, esquemas de codificación de red, pueden ser testeados y validados en la red desplegada, mediante la reprogramación de los nodos con las imágenes de código que realicen la funcionalidad correspondiente sobre los mismos.

Con esta finalidad, se ha implementado un protocolo para la programación vía radio, denominado MOTAP [Galache13a], tanto para nodos situados a un salto de distancia, como para aquellos a múltiples saltos de distancia. El protocolo permite la reprogramación de manera *unicast* (un sólo nodo), *multicast* (un conjunto de nodos) o *broadcast* (todos los nodos), tantas veces y con tantos códigos como sean necesarios.

4.1.2 Estado del arte

Como se puede derivar de los análisis realizados en [Brown06] y [Wang 06], los protocolos de reprogramación remota se pueden dividir en dos categorías principales, por un lado aquéllos que persiguen la reprogramación completa del código, ideados para nodos que pueden desempeñar diferentes funcionalidades y, por otro, los de reprogramación parcial de código, principalmente orientados a nodos con una funcionalidad específica que puede variar de manera parcial o para actualizaciones de código/firmware del nodo.

Dentro de los protocolos de reprogramación de código completo, el protocolo XNP [Jeong03] fue el primer protocolo para reprogramación de redes de sensores propuesto por *TinyOS* [TinyOS]. Este protocolo sólo permite la reprogramación a un salto de distancia (visión directa entre nodos), y no soporta la actualización incremental de la imagen cargada en el nodo. Para soportar la programación a múltiples saltos, se desarrolló el protocolo MOAP (*Multihop Over-the-Air Programming*) [Stathopoulos03], el cuál implementa un sencillo esquema de seguimiento de retransmisión fijando una ventana específica. El protocolo *Infuse* [Kulkarni04] presenta un esquema basado en TDMA (*Time Division Multiple Access*) para el envío de datos en forma de ráfaga en redes de sensores conocedoras de la ubicación. Para ello los nodos seleccionan periódicamente los predecesores y sucesores de la transmisión, apagando el receptor para las ranuras TDMA en los que no intervenga, con el consiguiente ahorro de consumo de energía y la reducción de las interferencias. En [Beutel04], se presenta una alternativa al acceso a los nodos a través de la red de sensores, proveyendo un plano de control paralelo que permita la actualización de los códigos de una manera más eficiente. Este plano de control se sustenta sobre una red que agrupa a varios de los nodos que se reprograman bajo un nodo controlador, permitiendo una gestión de la red más sencilla y eficaz.

El protocolo MNP (*Multihop Network Programming*) [Kulkami05] presenta un enfoque eficiente en términos de energía, basado en reducir los problemas asociados a las colisiones y al terminal oculto, buscando garantizar que dentro del área de cobertura de un nodo hay, a lo sumo, un nodo fuente transmitiendo el programa en un determinado momento. En una línea similar a MNP, se encuentra *Sprinkler* [Naik05], que presenta una cuadrícula virtual sobre la red física, computando un conjunto de dispositivos dominantes que evita las transmisiones redundantes, así como un orden de transmisión para evitar las colisiones. Por su parte, *Firecracker* [Levis04] presenta un enfoque en el que el servidor envía el código correspondiente a los nodos más alejados de la red, de forma que una vez que estos han recibido el código lo reenvíen al resto de los nodos. Al ser nodos que se encuentran distanciados entre sí, la probabilidad de colisión en este segundo reenvío del código disminuye considerablemente.

Se han implementado protocolos más complejos y eficientes, tales como *SYNAPSE* [Rossi08] y *Deluge* [Hui04]. El primero incorpora técnicas de codificación de canal conocidas como *Digital Fountain* [Luby97],[Luby01] embebidas en una solución ARQ (*Automatic Repeat Request*) híbrida (*Hybrid* ARQ), donde los datos se codifican previamente a la transmisión, y la recuperación frente a pérdidas se lleva a cabo mediante mecanismos de redundancia incremental, lo cual reduce considerablemente la sobrecarga en la transmisión. *Deluge* se basa en un protocolo consciente de la densidad de nodos de la red y con un mantenimiento de red epidémico para propagar datos de gran tamaño de uno o varios nodos fuente a muchos otros en base a comunicaciones a múltiples saltos. Para ello, las imágenes del código a transmitir se dividen en páginas de tamaño fijo, las cuales a su vez se dividen

en paquetes de menor tamaño. Esto permite paralelizar los envíos de diferentes páginas entre distintos nodos, así como evitar la fragmentación en la transmisión de los diferentes paquetes de código. Como mejora de este protocolo, en [Hagedorn08], se define el denominado *Rateless Deluge*, el cuál modifica el esquema de envío clásico de *Deluge* por uno de baja tasa de envío (*rateless*), utilizando combinaciones lineales de los *k* paquetes constituyendo cada una de las páginas, de forma que los nodos receptores puedan conformar el contenido de la página correspondiente mediante la recepción y procesamiento de *k* combinaciones lineales de los paquetes asociados a cada una de las páginas. De esta forma, para cada página de código, la fuente enviará combinaciones lineales de los paquetes que la conforman, hasta que reciba la confirmación de que todos los nodos han decodificado correctamente esta página. Finalmente, en [Law11] se define el protocolo *SReluge* (*Secure Rateless Deluge*), que añade una capa de seguridad al protocolo *Rateless Deluge*, contra los ataques de activos caracterizados por perseguir la interrupción del servicio (en este caso el proceso de reprogramación), con el envío masivo de paquetes codificados no válidos. Para solventar este problema, *SReluge* utiliza un sistema de clasificación de vecinos para aislar a los nodos generadores de paquetes no válidos, así como una técnica combinatoria para decodificar paquetes en la presencia de nodos espurios.

En lo referente a los mecanismos de reprogramación de código parcial, [Reijers03] presenta un esquema de distribución de código diferencial que genera y propaga a los nodos, en lugar de la imagen completa. Esta distribución es muy eficiente en términos de energía para actualizar de manera inalámbrica el código ejecutándose en los nodos de una red de sensores, puesto que permite actualizar sólo aquellas partes del código que se han modificado, asignadas a determinadas partes físicas de la memoria que se sobrescriben con el correspondiente código. *Rsync* [Jeong04] se erige como un algoritmo dirigido a las actualizaciones de *software* incrementales, diferenciándose de *Reijers* en que es independiente del procesador y en que no requiere del conocimiento de la estructura del código. En este sentido, el servidor envía un esquema diferencial con los cambios referentes a la nueva actualización de código y el cliente responde con aquellos bloques que cambian respecto a su imagen de código actual, de forma que el servidor se los pueda enviar. En [Koshy05] se presenta un sistema de actualización, tanto estática como dinámica, de tipo modular, en el que las diferentes funciones que constituyen el código a ejecutar, reservan un espacio al final de las mismas para permitir futuras expansiones. De esta forma las funciones pueden ser actualizadas de manera independiente mediante el envío de códigos diferenciales.

A continuación, derivado de los diferentes protocolos y algoritmos definidos, así como teniendo en consideración las características inherentes al despliegue, se detalla la implementación del protocolo de reprogramación remota realizado en el marco de este trabajo.

4.2 IMPLEMENTACIÓN DEL PROTOCOLO

Los casos de uso (descritos en el Capítulo 3) en los que se centra esta Tesis Doctoral son aquéllos en los que se permite la experimentación a nivel de nodo, lo que se traduce en la necesidad de poder reprogramar los nodos de manera remota. En este sentido, dentro de los casos de uso previamente comentados, se incluyen tanto nodos fijos como nodos móviles, donde todo el tráfico referente a la reprogramación (gestión de red), se realizará a través de las interfaces *Digimesh* y *GPRS*, respectivamente. Todos estos nodos están provistos por la empresa Libelium, que ofrece su propia

solución para la reprogramación vía radio aunque, de la misma forma que ocurre con la mayoría de las soluciones existentes, se encuentra principalmente limitada a despliegues de pequeño tamaño y con entornos radioeléctricos poco hostiles. En este sentido, para adaptar el funcionamiento del protocolo al despliegue llevado a cabo así como para hacerlo compatible con la arquitectura de SmartSantander, se ha adaptado el funcionamiento de aquél, haciéndolo más robusto y eficiente. Dentro de su implementación se pueden diferenciar tres partes fundamentales: el servidor de comunicaciones, el servidor OTAP y el cliente OTAP.

En este capítulo se especificará el funcionamiento del protocolo para los nodos fijos, sobre los que se van a llevar a cabo las medidas para la caracterización del mismo; aunque también se realizará una reseña de la forma en que se adapta el citado protocolo para el funcionamiento con los nodos móviles.

4.2.1 Servidor de comunicaciones

Conviene recordar aquí que el servidor de comunicaciones se erige principalmente en un multiplexor de puertos que ofrece varios puertos de comunicación virtuales hacia un único puerto físico.

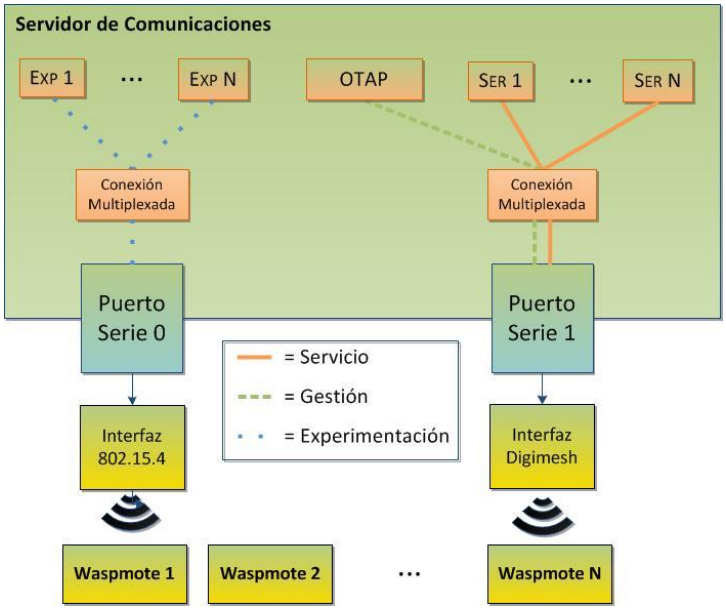

Figura 4.1. Arquitectura del servidor de comunicaciones

En la figura 4.1, se puede observar que para todo el tráfico asociado al OTAP, se crea el correspondiente puerto virtual (denominado OTAP) a través del cual se llevará a cabo la transmisión/recepción de todos los paquetes de datos asociados al proceso de reprogramación. El puerto físico sobre el que se realiza el OTAP es el mismo (interfaz *Digimesh*) por el que se transmiten los datos asociados a los diferentes servicios desplegados. De esta forma, en el puerto físico de *Digimesh* se recibirán datos asociados, tanto al servicio como a la reprogramación, siendo enviados a los correspondientes puertos virtuales. Para llevar a cabo esta clasificación, el servidor de comunicaciones analiza la trama recibida y, en función del identificador de la misma, la reenvía hacia el puerto virtual correspondiente.

4.2.2 Servidor y cliente OTAP

Para llevar a cabo la carga de un programa a través del OTAP, se ha de instalar una instancia a nivel de servidor y otra a nivel de cliente. Desde el punto de vista del servidor, se implementa el código correspondiente en todas las pasarelas desplegadas, donde se almacenan las correspondientes imágenes a ser enviadas hacia los diferentes nodos que dependen de la pasarela correspondiente. Respecto al lado del cliente, la instancia para realizar el OTAP, es implementada en todos los repetidores desplegados, adaptada a las restricciones en términos de memoria y de procesador asociadas a estos nodos. Teniendo en cuenta que el *waspmote* implementa un procesador mono-hilo, se utilizan las interrupciones del microcontrolador asociadas al *watchdog* (esquema mostrado en el Capítulo 3), de forma que se chequea de manera periódica la recepción de un determinado comando (en este caso el comienzo del proceso de reprogramación) en la interfaz *Digimesh*, al mismo tiempo que continúan enviándose las tramas asociadas al servicio, así como las relacionadas con la experimentación a nivel de nodo. Es importante destacar que, una vez que el nodo entra en el proceso de reprogramación, se interrumpen tanto el servicio como la experimentación, hasta que el proceso de programación haya terminado. Sin embargo, si el nodo no es el objetivo de la reprogramación y sólo se comporta como un puro retransmisor de la información (forma parte de la ruta hacía un nodo que si va a ser reprogramado), entonces tanto la gestión de red, la provisión de servicio como la experimentación realizada por ese nodo se mantiene sin interrumpirse, pudiendo sufrir algún retraso/pérdida de paquetes por la alta tasa de datos que pasa a través del nodo). Los servicios de monitorización medioambiental fija y riego inteligente tienen una tasa de envío muy baja (cada 5 minutos), y no sufren ninguna alteración por encontrarse involucrados en un proceso de reprogramación.

Todos los paquetes involucrados en el proceso de MOTAP, se transmiten/reciben a través del puerto virtual OTAP ofrecido por el servidor de comunicaciones y, además, todos ellos llevarán incluida una cadena de 8 octetos de tamaño (clave de reprogramación), que permite al nodo receptor identificar aquellos paquetes asociados a un proceso de reprogramación. En la figura 4.3 y la figura 4.4, se muestran la secuencia de reprogramación, tanto en modo *unicast*, como *multicast*/*broadcast*, respectivamente; detallándose a continuación los pasos realizados por ambos procesos de reprogramación:

- El proceso de MOTAP comienza con una trama de inicio, (en la que se incluye la correspondiente clave de reprogramación), para informar al nodo (reprogramación *unicast*), o a los nodos (proceso *multicast*/*broadcast*) objeto del proceso de reprogramación, del comienzo del mismo. Este paquete se envía en modo *unicast* para transmisiones *unicast* y, en modo *broadcast*, para reprogramaciones de tipo *multicast* y *broadcast*, incluyendo los siguientes campos:

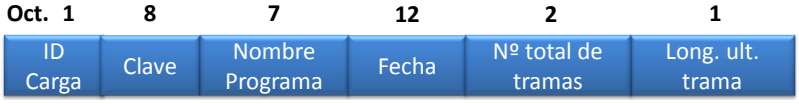

Figura 4.2. Trama de inicio de reprogramación

  Según se muestra en la figura 4.2, la trama correspondiente posee el *nombre* y la *fecha* con los que el programa se va a almacenar en el nodo, el *número total de fragmentos* de longitud fija, así como la *longitud de la última trama*, información necesaria ya que el tamaño del código no tiene que corresponderse con un múltiplo exacto de la longitud fija de fragmento. Al principio de cada

trama, se incluye el *ID de la carga* para que el paquete sea reconocido en el destino y tratado adecuadamente. Para que el nodo receptor procese correctamente la trama y la asocie a la reprogramación, se incluye el campo de la *Clave* dentro del paquete.

Se ha de precisar que para comunicaciones *multicast*, con la finalidad de no interrumpir la provisión de servicio ni la experimentación asociadas a los nodos no afectados por el proceso de reprogramación, se envía (en modo *unicast*) una trama (anterior a la de inicio) a los nodos a ser reprogramados, para modificar la clave OTAP utilizada para reprogramarlos. De esta forma, con esta nueva clave (no conocida ni asociada por el resto de nodos a un proceso de reprogramación), sólo estos nodos interrumpirán la provisión de servicio y la experimentación, manteniendo el resto de nodos del *cluster* su funcionalidad de manera intacta, como ya se comentó con anterioridad. Para comunicaciones *broadcast*, no será necesario enviar este primer paquete puesto que todos los nodos se encuentran incluidos como destinatarios del proceso de reprogramación. El envío de este paquete es la única diferencia entre las reprogramaciones *multicast* y *broadcast*, ya que el envío *multicast* se corresponde con una transmisión tipo *broadcast* que sólo procesan los nodos que han sido notificados inicialmente con el paquete descrito.

Desde el punto de vista del cliente, el proceso de MOTAP comienza cuando el nodo recibe la trama de inicio, entrando éste en el proceso de reprogramación e interrumpiéndose tanto la provisión de servicio, como la experimentación, que se estén llevando a cabo en el mismo. Una vez recibida esta trama de inicio, el nodo notifica al servidor que ha entrado en el estado de reprogramación, mediante el envío de un paquete de confirmación (su estructura se indica en la ***figura 4.7*** y se detalla más adelante). Como se puede observar en la figura 4.3 y la figura 4.4, esta confirmación se envía siempre en modo *unicast*, al igual que el resto de mensajes enviados desde los nodos al servidor dentro del proceso de reprogramación.

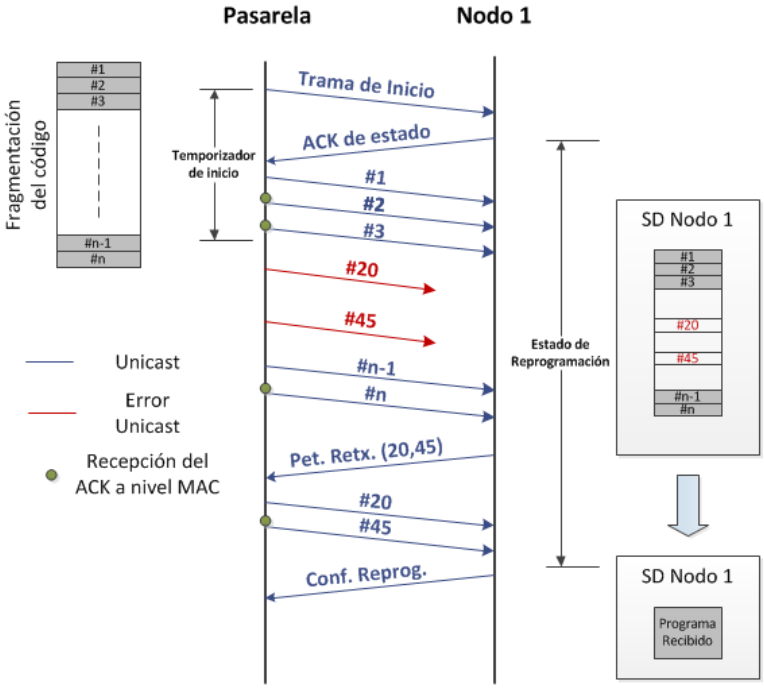

Figura 4.3. Proceso OTAP unicast

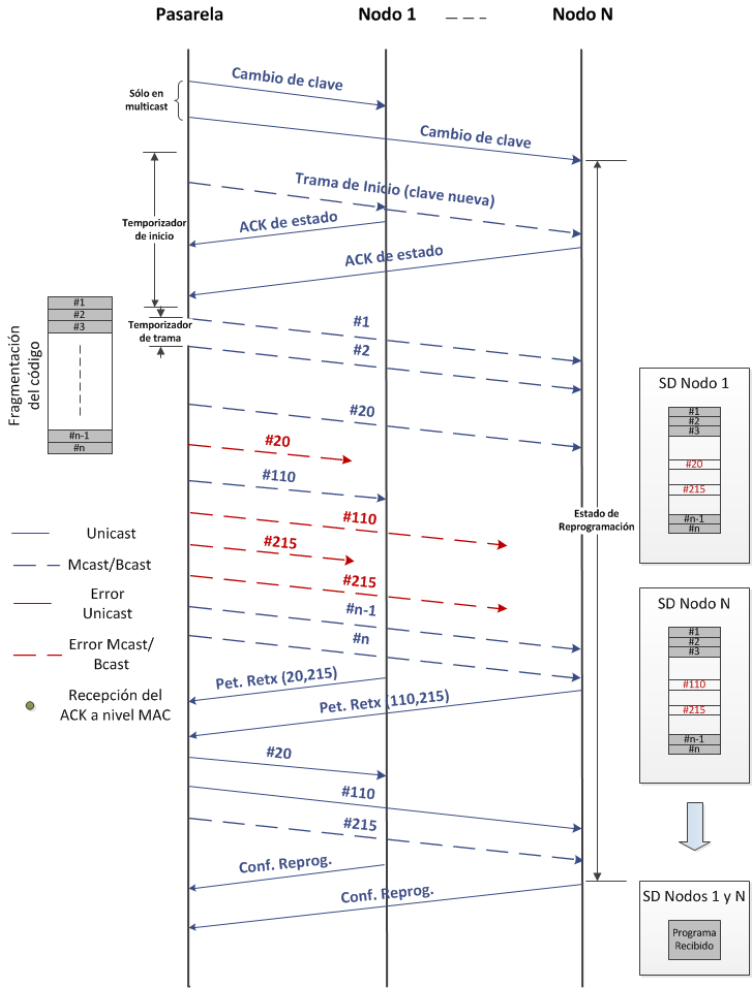

Figura 4.4. Proceso OTAP multicast/broadcast

- Una vez recibida la confirmación de que el nodo ha entrado en estado de reprogramación, el servidor divide la imagen de código completa en fragmentos de tamaño fijo, enviando el número correspondiente de tramas atendiendo a la siguiente estructura de paquete:

Figura 4.5. Trama de datos de reprogramación

Como se puede observar en la figura 4.5, la estructura de trama está compuesta del *número de fragmento* y de los *datos* asociados a ese fragmento correspondiente que tiene una longitud fija de 60 octetos (en el apartado de medidas se explicará el valor de esta longitud fija de paquete). Por supuesto, como es un paquete de reprogramación, debe estar incluida la *clave de reprogramación* para que éste sólo sea procesado por los nodos objetivo de la reprogramación. Al igual que el paquete de inicio al comienzo de esta trama se indica el *ID de la carga*, para que el paquete se procese convenientemente en el nodo destino.

Teniendo en consideración las grandes restricciones en términos de memoria interna (128KB), y que el tamaño medio de un código ejecutándose en el *waspmote* ocupa entre 50KB y 120KB (depende de las funcionalidades y librerías utilizadas), el código enviado debe ser almacenado en una memoria adicional. Con esta finalidad, el *waspmote* posee una ranura para introducir una

memoria SD externa de 2 GB para almacenamiento adicional, donde se guardan los diferentes códigos recibidos.

- El servidor envía fragmentos separados un determinado intervalo de tiempo, cuyo valor dependerá del tipo de proceso de reprogramación, bien sea *unicast* o *multicast/broadcast*. En el caso *unicast*, el servidor envía un paquete y espera el reconocimiento a nivel MAC (ofrecido por *Digimesh*) de que ha sido recibido por el destinatario, enviando el paquete siguiente. El número de retransmisiones a nivel MAC, se fija en el módulo *DIgimesh* y en el apartado de medidas se indicará su valor concreto. Respecto a las transmisiones *multicast/broadcast*, los nodos no envían reconocimiento a nivel MAC (no habilitado el módulo Digi), ya que inundaría la red y retrasaría el proceso si hubiera que esperar a todos los reconocimientos y, a nivel de servidor, éste sólo recibirá la confirmación de su propio módulo *Digimesh* de que el paquete ha sido enviado. En este sentido, se ha de fijar por lo tanto, un tiempo mínimo medio entre tramas en la fuente (valor definido en el apartado de medidas). Si la confirmación a nivel MAC del módulo *Digimesh* del servidor no se recibe dentro de este periodo, el servidor esperará hasta la correspondiente recepción de este paquete de confirmación, dentro de un tiempo máximo fijado (tiempo de ruta perdida), superior al tiempo entre tramas. Si este paquete no es recibido, se genera un error en el descubrimiento de la ruta hacia ese nodo, el servidor retransmitirá de nuevo (hasta un máximo) el correspondiente fragmento, un determinado número de veces a nivel de aplicación hasta que, si no se recibe el reconocimiento, se retransmita el siguiente paquete. Por último, cabe destacar que, en las transmisiones *unicast*, si la pérdida de paquetes (principalmente por caída de la ruta, desconexión momentánea del módulo *Digimesh*), es superior a un determinado valor (indicado también en el apartado de medidas), la fuente comenzará a enviar de nuevo desde el último paquete que no se envió correctamente.

  Desde el punto de vista del nodo, una vez que recibe un fragmento, lo coloca en la posición correspondiente de la memoria SD. Esta posición es conocida puesto que la ubicación del fichero donde se almacenan las tramas queda determinada cuando éste es creado al comienzo de la reprogramación, y teniendo en consideración el índice de cada uno de los fragmentos y su longitud fija, se puede determinar la posición de memoria donde ha de ubicarse cada uno de ellos. Esto permite a la fuente transmitir en modo continuo, sin tener que detener la transmisión para realizar la retransmisión de paquetes perdidos (salvo el caso citado de que se pierdan un gran número). Este modo de operación es muy importante en las comunicaciones *multicast*/*broadcast*, puesto que un nodo puede perder un fragmento, pero el resto de los nodos pueden haberlo recibido correctamente, pudiendo éstos recibir este fragmento por duplicado si el nodo fuente tuviera que retransmitirlo, y acarreando la consiguiente disminución del rendimiento del protocolo.

  Como se puede observar en la figura 4.3, el nodo 1 ha perdido los fragmentos número 20 y 45, mientras que en la figura 4.4, el Nodo 1 ha perdido los fragmentos número 20 y 215, y el nodo N los fragmentos 110 y 215.
- El nodo almacena el identificador asociado a los fragmentos perdidos en una lista, lo que permite pedir la retransmisión de los mismos cuando la transmisión de todos los fragmentos haya finalizado. Cuando el último fragmento de código (el nodo conoce el índice del último fragmento, puesto que recibió el valor del número total de fragmentos en la trama de inicio) es recibido, el nodo chequea la lista que contiene los fragmentos perdidos y pide la retransmisión de los mismos a la fuente. La estructura del paquete correspondiente, se muestra a continuación:

Figura 4.6. Trama de petición de retransmisión de paquetes perdidos

Como se muestra en la figura 4.6, la trama de retransmisión está compuesta por un campo incluyendo el *número total de tramas perdidas*, así como el *índice de las tramas* que se han perdido. Si un nodo pierde un porcentaje de tramas elevado, considera que la transmisión ha sido errónea, saliendo del modo de reprogramación y enviando el correspondiente paquete de estado (se muestra figura 4.7 y se detalla más adelante más adelante) indicando que el proceso de reprogramación se ha detenido en el nodo. Si el proceso de reprogramación es de tipo *unicast* el servidor detendrá el envío de tramas hacia el nodo, pero si es *multicast*/*broadcast* el servidor seguirá enviando tramas al resto de nodos. Asimismo, el nodo tiene un temporizador que inicia al comienzo de la reprogramación y que, si expira, antes de que el proceso finalice, el nodo considera errónea la reprogramación, enviando el correspondiente paquete de estado (figura 4.7). Cabe destacar que esta trama no lleva la clave de reprogramación, puesto que es enviada de los nodos a la pasarela (ésta siempre procesa las tramas que recibe) y, por lo tanto, con el *identificador de carga* es suficiente para llevar a cabo el correspondiente procesado de la trama.
En el caso de la figura 4.3, el nodo 1 enviará un paquete con los fragmentos perdidos número 20 y 45, mientras que en la figura 4.4, el nodo 1 enviará los fragmentos perdidos 20 y 215, y el nodo N el 110 y el 215.

- Como se ha indicado en el punto anterior, el servidor no realiza las retransmisiones de los paquetes perdidos (requeridas por los nodos de destino), durante el proceso de reprogramación, sino que cuando ha finalizado la transmisión de todos los fragmentos, el servidor recibirá la lista de los que se han perdido por cada uno de los nodos de destino. De esta forma, para transmisiones *unicast*, el servidor enviará en modo *unicast* todos los paquetes que aparecen en la lista al nodo correspondiente, según se muestra en la figura 4.3 donde reenvía los paquetes 20 y 45. Sin embargo, para el caso de transmisiones *multicast/broadcast*, el servidor procesa las listas de fragmentos recibidas de cada uno de los nodos, reenviando en modo *unicast* aquellos fragmentos solicitados por un solo nodo (paquete 20 al nodo 1 y paquete 110 al nodo N), y en modo *broadcast* aquellos fragmentos pedidos por más de un nodo (fragmento 215).

  Como se ha apuntado durante la explicación de los diferentes pasos del proceso de reprogramación, a continuación se muestra el paquete de estado que se utiliza para comunicar las distintas notificaciones enviadas desde los nodos al servidor (excepción hecha del paquete de petición de fragmentos perdidos mostrado en la figura 4.6).

Figura 4.7. Trama de notificaciones de estado

Así, la figura 4.7 muestra la trama de estado que se compone del correspondiente *ID carga* para que sea procesada de forma acorde en el servidor, así como el *ID Origen* que representa el *ID de la carga* a la que esta trama de estado responde. Respecto al campo de estado, éste puede tomar diferentes valores atendiendo a distintas incidencias, como se describe a continuación:

- Entrada en proceso de reprogramación: El nodo envía una trama de estado en respuesta a la trama de inicio enviada por el servidor, indicando que ha entrado en proceso de reprogramación y que está preparado para recibir tramas de datos de reprogramación.
- Acceso erróneo a la tarjeta SD. El nodo no puede abrir correctamente la tarjeta SD para crear el fichero correspondiente o se produce un error cuando se quieren guardar los fragmentos en la tarjeta SD.
- Tarjeta SD en uso: El nodo intenta acceder a la memoria SD, pero ésta se encuentra utilizada por otra aplicación ejecutándose en el nodo (ya sea experimentación o servicio), de forma que se lanza un mensaje de error deteniéndose el proceso de reprogramación correspondiente.
- Pérdida de ruta. Debido a la caída de uno de los nodos intermedios que forman la ruta desde el servidor a cada uno de los nodos de destino, se produce la pérdida de la ruta correspondiente y, por lo tanto, la pérdida de un gran número de paquetes asociados a la reprogramación.
- Apagado del módulo radio. En alguno de los nodos destino, puede suceder la desconexión del módulo radio de forma momentánea, produciendo que el servidor entienda que se ha perdido la conexión con este nodo y, por lo tanto, se detenga el envío del código correspondiente.
- Excedido el tiempo máximo de respuesta. El servidor espera un tiempo determinado para la recepción del mensaje de confirmación de los nodos, de forma que si se excede este tiempo máximo se cancela el proceso de reprogramación hacia ese nodo.
- Archivo ya existente. Los archivos que se envían a un determinado nodo se almacenan en la SD con un nombre (fijado por el usuario) de 7 octetos de tamaño. Una vez que se quiere llevar a cabo la carga del programa, si el nombre elegido para almacenarlo ya existe, entonces se detiene el proceso de reprogramación, indicando al usuario la existencia de otro programa almacenado en la SD con ese mismo nombre.
- Elevado número de paquetes perdidos en una transmisión: El nodo receptor detecta que se han perdido excesivos paquetes en una transmisión, lo que asocia a una baja calidad de la ruta de comunicaciones y, por lo tanto, se envía un mensaje de error para detener la reprogramación del correspondiente nodo.

Hay que tener en cuenta que todos los errores indicados se encuentran asociados a comunicaciones *unicast*, puesto que en aquellas *multicast/broadcast* el servidor no debe detener el proceso de reprogramación porque en alguno de los nodos se produzca uno de los errores anteriormente citados, ya que el resto puede estar recibiendo el código correctamente. Una vez terminado el proceso de reprogramación *multicast/broadcast*, el servidor es notificado de aquellos que se reprogramaron de manera correcta, de forma que para el resto de nodos que no han finalizado correctamente el proceso de reprogramación, el servidor deberá lanzarlo nuevamente hacia ese conjunto de nodos (el proceso de reprogramación no terminó por alguno de los errores previamente citados), o reenviar las tramas requeridas por cada uno de los nodos.

4.3 MEJORAS DEL PROTOCOLO DE REPROGRAMACIÓN

Aunque desde el punto de vista de envío de las imágenes de código (dividiendo el fichero completo a enviar en tramas de longitud fija), la implementación realizada funciona de manera similar a la de Libelium y a algunas de las que se mostraron en el estado del arte (asociadas a la reprogramación de

código completo), en lo referente a la recepción, procesado y almacenamiento de estos fragmentos se consignan las siguientes diferencias y características (algunas asociadas al despliegue específico):

- Provisión de servicio y capacidad de experimentación de manera simultánea a la programación remota (OTAP), en el nodo servidor (pasarela): El servidor de comunicaciones permite que la pasarela pueda reprogramar un conjunto de nodos dentro de su *cluster*, al mismo tiempo que continúa recibiendo datos referentes al servicio y a la experimentación, generados por el resto de nodos del *cluster* que no intervienen en el proceso de reprogramación. Este funcionamiento no era posible en la implementación de partida, en la que la realización de un proceso de reprogramación sobre uno o varios nodos, se traducía en la interrupción de la recepción de los datos de experimentación y servicio de todos los nodos del *cluster*.
- Gestión de la reprogramación remota (OTAP) y los datos de experimentación/servicio de manera concurrente en el cliente: Debido a las limitaciones en términos de memoria y procesador, así como por la ausencia de un sistema operativo, no es posible implementar en los *waspmotes* una instancia del servidor de comunicaciones que se encuentra instalado en las pasarelas. En este sentido, para intentar mantener una ejecución cuasi-concurrente de la experimentación, el servicio y la gestión a nivel de nodo, y considerando las limitaciones asociados al procesador mono hilo del que se encuentra provisto el *waspmote*, se utilizan las interrupciones provistas por el mecanismo de *watchdog.* La finalidad de estas interrupciones consiste en chequear periódicamente la interfaz *Digimesh* para gestionar los paquetes referentes al proceso de OTAP, así como para enviar los correspondientes datos de servicio; realizando el resto de tiempo el envío de las tramas asociadas a los diferentes experimentos que se llevan a cabo en los nodos. La independencia física, tanto a nivel de servidor como de nodo, se puede asimilar con el enfoque ofrecido por [Beutel04], enviándose en este caso los datos de experimentación a través de un plano físico distinto respecto al de los datos de gestión.
- Gestión de paquetes desordenados: El almacenamiento de los fragmentos recibidos en posiciones conocidas de la memoria SD de los nodos destino, permite la gestión de paquetes recibidos fuera de orden, principalmente asociados a los procesos *multicast/broadcast* a múltiples saltos donde las variaciones del entorno pueden conllevar cambio en la asignación de las rutas, provocando la posible llegada de paquetes fuera de orden. Esta nueva gestión de los paquetes recibidos fuera de orden, mejora la anterior gestión de paquetes desordenados, en la cual se almacenaban los dos últimos paquetes recibidos en la memoria *flash* del dispositivo, de forma que sólo se tenía la capacidad de resolver la llegada fuera de orden de una posición, significando la llegada de un paquete dos o más posiciones fuera de orden sobre su índice correspondiente, el fallo del proceso de reprogramación. A diferencia de otros protocolos como MOAP que implementan técnicas de ventana deslizante para evitar la llegada de paquetes fuera de orden, en este caso el protocolo presentado es capaz de gestionar estos paquetes recibidos fuera de orden sin necesidad de enviar reconocimiento de los paquetes recibidos ni de, en casos de congestión, detener la transmisión, puesto que se conoce la posición en la que el paquete perdido ha de almacenarse.
- Mecanismos de retransmisión de paquetes perdidos: La implementación inicial, bajo la suposición de entornos de trabajo controlados, confiables , poco interferentes y con un tamaño acotado, no proveía ningún mecanismo de retransmisión a nivel de aplicación, confiando en los mecanismos de retransmisión implementados a nivel MAC por los módulos XBee. Lógicamente, el entorno urbano en el que trabaja esta implementación presenta unas características

absolutamente diferentes (entorno interferente), que requieren de mecanismos de retransmisión adicionales a nivel de aplicación, para asegurar la correcta realización del proceso de reprogramación. Con esta finalidad, los nodos que intervienen en el proceso de reprogramación almacenan los identificadores de los fragmentos que no han recibido, pidiendo la consiguiente retransmisión de los mismos por parte de la fuente al final del proceso de reprogramación. El hecho de pedir la retransmisión de los paquetes al final del proceso de reprogramación permite aumentar el rendimiento del proceso, puesto que algunos de los paquetes perdidos pueden ser enviados de manera *multicast*, si hubieran sido pedidos por varios de los nodos participantes en el proceso. Por otro lado, la capacidad de gestionar los paquetes que llegan de manera desordenada (explicado en el punto anterior), permite que se pueda realizar de esta forma el proceso de retransmisión, puesto que de lo contrario habría que implementar mecanismos de parada y espera o de ventana deslizante, que conllevarían la retransmisión en tiempo de ejecución en lugar de al final del proceso. Este procedimiento de negociación entre los nodos clientes y el servidor es similar a los procedimientos utilizados por MNP y Deluge, para evitar inundar la red con envíos redundantes.

Como se puede observar, las principales mejoras del protocolo descrito radican principalmente en una gestión más eficiente de paquetes desordenados y perdidos, consiguiendo un mejor rendimiento en entornos interferentes; así como la adaptación a las particularidades del despliegue llevado a cabo en Santander donde la gestión de red, en general, y la reprogramación remota, en este caso particular, deben coexistir de manera simultánea y concurrente con los tráficos asociados al servicio provisto y a la experimentación realizada sobre cada nodo.

El protocolo que se ha implementado se encuentra claramente orientado hacia la reprogramación de código completo, puesto que su finalidad es la de cargar diferentes experimentos (en principio totalmente diferentes entre ellos) en los nodos, requiriendo una reprogramación completa de los mismos. Sin embargo, es importante reseñar que, desde el punto de vista de gestión de la red, para actualizaciones de la imagen por defecto cargada en los nodos, resultaría interesante incluir una funcionalidad para actualizar el código existente, modificando en el fichero guardado en la SD, sólo aquellas líneas que hubiesen variado, basándose en los conceptos de alguno de los algoritmos diferenciales descritos en el apartado del estado del arte.

4.4 FUNCIONALIDADES AÑADIDAS

El protocolo de OTAP, además de desarrollar la funcionalidad para el envío de diferentes códigos desde la pasarela a los diferentes nodos desplegados mediante el proceso descrito en el apartado anterior, también implementa otros comandos relacionados con la reprogramación, que se definen a continuación:

- *Reset*: Este comando ofrece la capacidad de reiniciar los nodos de manera remota, permitiendo recuperarlos de un funcionamiento anómalo y volviendo a iniciarlos con la última imagen que estuvieran ejecutando o con la imagen por defecto. Para indicar que el proceso se ha realizado correctamente, el nodo devuelve el correspondiente mensaje de confirmación.
- *Start_new_program*: Como se ha indicado, los programas enviados desde el servidor a cada uno de los nodos, se almacenan en la tarjeta SD externa de la que éstos se encuentran provistos. De

entre todos estos programas, el comando *start_new_program* permite seleccionar uno de ellos, para que el nodo se reinicie y empiece a ejecutarlo. Si la operación se realiza correctamente, el nodo envía el correspondiente mensaje indicando que se está ejecutando el nuevo código. Sin embargo, si por algún motivo la carga no se produjera correctamente (por ejemplo, error en el acceso a la SD o en la carga del código correspondiente en el *bootloader*), el nodo indicará que se sigue ejecutando el código anterior.

- *Get_boot_list*: La ejecución de este comando devuelve los diferentes programas que se encuentran almacenados en la memoria SD de un determinado nodo.
- *Scan*: Comando utilizado para comprobar que un nodo se encuentra operativo, devolviendo además el programa correspondiente que se está ejecutando en el nodo en ese momento.
- *Delete_program*: Para evitar una acumulación excesiva de programas dentro de la tarjeta SD del nodo, este comando permite eliminar aquellos que se consideren oportunos. Aunque el tamaño de la SD permite el almacenamiento de miles de programas, en aras de aligerar la respuesta del nodo al comando *get_boot_list*, así como para ofrecer espacio libre en la SD para almacenamiento de datos asociados a los diferentes servicios/experimentos realizados sobre el nodo, es importante eliminar aquellos programas que no se necesiten en el nodo (versiones anteriores de código, experimentos ya realizados).

La ejecución de todos estos comandos, unida a la del de reprogramación anteriormente descrito, permiten la gestión de los nodos desde el punto de vista del control de los programas que se están ejecutando y almacenando en el mismo, otorgando de una gran flexibilidad a la red para cambiar el comportamiento de los nodos que la forman, ya sea a nivel de servicio o de experimentación. Todos estos comandos pueden ser ejecutados a nivel tanto *unicast* como *multicast*/*broadcast*.

Es importante resaltar que los comandos anteriormente descritos, a nivel funcional son herencia de la implementación realizada por Libelium, adaptándose a nivel operacional para trabajar correctamente con la nueva implementación, con la finalidad de poder permitir el envío/recepción de estos mensajes manteniendo de manera simultánea la provisión de servicio, así como la experimentación, como ya se explicó en apartados anteriores.

4.5 Caracterización de la implementación: Medidas y resultados

Para llevar a cabo la caracterización y validación del protocolo de reprogramación remota implementado en este capítulo, se realizan diferentes medidas, tanto en el banco de pruebas de interiores como en una parte del despliegue en el exterior (*cluster* 6), comparando los resultados correspondientes obtenidos.

Figura 4.8. Escenario de exteriores (Cluster 6)

En la figura 4.8 se muestra el escenario de exteriores compuesto por los 26 nodos pertenecientes al *cluster* de la pasarela 6 (instalado en el centro de la ciudad); mientras que en la ***figura 4.9*** aparece el banco de pruebas en interiores que agrupa 15 nodos ubicados en los techos de los diferentes despachos y pasillos de las instalaciones del Grupo de Ingeniería Telemática, en el edifico de los Laboratorios de I+D+i de Ingeniería de Telecomunicación). Todos estos nodos se encuentran conectados mediante cables USB que, por un lado, alimentan las baterías recargables y por el otro, permiten la depuración de la operación de los nodos, a través de la mencionada interfaz cableada.

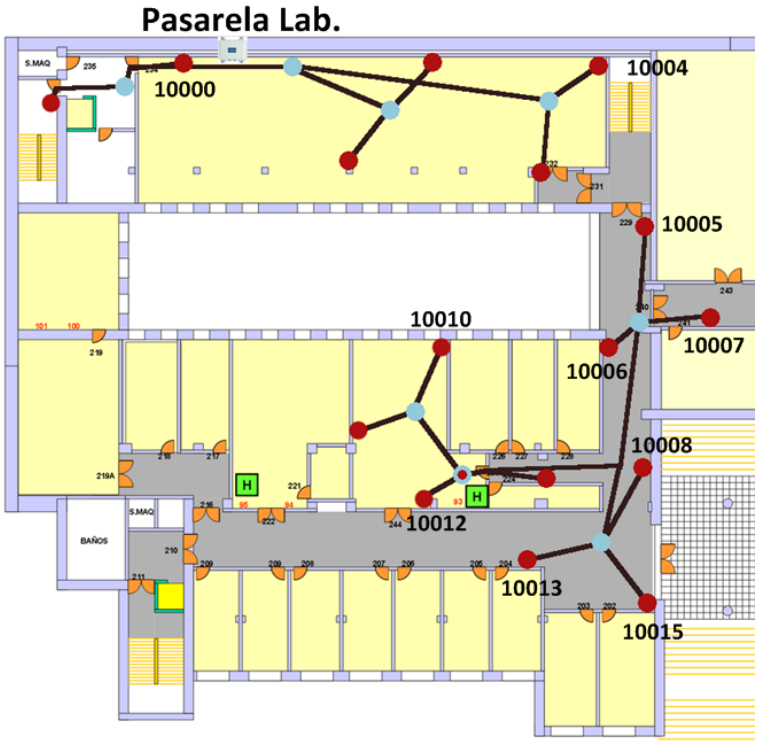

Figura 4.9. Escenario en interiores (Laboratorio)

De acuerdo a las medidas a llevar a cabo, han de considerarse las siguientes condiciones de contorno, principalmente relacionadas con las características del estándar 802.15.4 [IEEE802.15.4], así como las particularidades del protocolo *Digimesh*, que se ejecuta sobre el estándar 802.15.4 nativo:

- En primer lugar, el estándar 802.15.4 presenta dos mecanismos de acceso al canal: *beacon* y *non-beacon*, ambos basados en acceso múltiple con escucha de portadora evitando colisiones (*Carrier Sense Multiple Access with Collision Avoidance, CSMA/CA*). El modo *beacon* requiere un coordinador de la red que, a intervalos regulares, envía mensajes de baliza para la sincronización y asociación de la red. Por su parte, el modo *non-beacon* se basa en un CSMA/CA no ranurado que no requiere de la transmisión de balizas, lo que se traduce en una comunicación completamente descentralizada entre los nodos, incluyendo además una sobrecarga menor que el modo *beacon*. Teniendo en cuenta que, tanto el enrutamiento como la organización de la red, serán realizadas por el protocolo *Digimesh*, así como la menor sobrecarga introducida por el modo *non-beacon*, los dispositivos trabajarán en este modo.
- Tasa de transmisión de datos: El máximo caudal eficaz ofrecido por el estándar 802.15.4 a nivel físico es de 250 Kbps, pero la velocidad máxima de los nodos (placa *waspmote*) es de 38.4 Kbps en el interfaz radio, lo que supone un cuello de botella para la comunicación cableada. Por este motivo, se utilizará también esta tasa de 38.4 Kbps para las comunicaciones en el puerto serie entre la pasarela y el módulo *Digimesh*.
- Tamaño de fragmento: Para determinar el tamaño máximo de fragmento utilizado dentro del protocolo, se debe partir en primer lugar de la trama 802.15.4 nativa a nivel MAC, que se muestra a continuación:

<table>
<tr><td>Octetos: 2</td><td>1</td><td>0/2</td><td>0/2/8</td><td>0/2</td><td>0/2/8</td><td>0/5/6/10/14</td><td>variable</td><td>2</td></tr>
<tr><td rowspan="2">Control trama</td><td rowspan="2">Número Secuencia</td><td>PAN ID Destino</td><td>Dirección Destino</td><td>PAN ID Fuente</td><td>Dirección Fuente</td><td rowspan="2">Cabecera Seguridad Auxiliar</td><td>Carga Útil</td><td rowspan="2">Cod. Verificación</td></tr>
<tr><td colspan="4">Casmpos de direccionamiento</td><td></td></tr>
<tr><td colspan="7">Cabecera MAC</td><td>Carga Útil MAC</td><td>Cola MAC</td></tr>
</table>

Figura 4.10. Formato general de la trama MAC 802.15.4 (Fte. [802.15.4])

La longitud total de la trama a nivel MAC (mostrada en la figura 4.10) es de 127 octetos, dividiéndose en los siguientes campos:

- Campos fijos: Los campos de *control de trama*, *número de secuencia* y la *suma de verificación*, son campos propios del protocolo que ocupan 5 octetos.
- Direccionamiento: En las redes 802.15.4, el direccionamiento de los nodos se puede realizar utilizando 16 bits (2 octetos) o 64 bits (8 octetos) y, adicionalmente, existen los denominados *identificadores de PAN* (PANID) fuente y destino, los cuales ocuparán 2 octetos para la fuente y 2 para el destino, siempre y cuando no exista un coordinador de PAN (trabajando en modo *beacon*), en cuyo caso tendrían tamaño 0. Considerando que se trabaja con un direccionamiento de 64 bits y en modo *non-beacon*, estos campos ocupan un tamaño de 20 octetos.
- *Cabecera de seguridad auxiliar*: Campo que tiene una longitud variable y sólo se utiliza si la seguridad está habilitada. Como en este caso no se encuentra habilitada, su valor será 0.

- *Carga útil de la trama*: Teniendo en cuenta la longitud de los campos previamente descritos (25 octetos), la carga útil de la trama MAC 802.15.4 es de 102 octetos. Adicionalmente, DIgi se reserva 2 octetos de esta carga útil para lo que denomina *Mac Mode* (MM), el cual introduce una cabecera adicional para comunicaciones. El valor de MM puede ser 0, 1 o 2, siendo sólo los modos de funcionamiento 1 y 2 estrictamente compatibles con 802.15.4. El modo 1 implica que no hay reconocimientos (ACKs) a nivel MAC ni para transmisiones *broadcast*, ni *unicast*; mientras que el modo 2 (modo fijado en los módulos *Digimesh*), implica un comportamiento estándar incluyendo reconocimientos para los paquetes *unicast*, pero no para los *broadcast* (como se indicó en la explicación del proceso de reprogramación). De esta forma, la carga útil de la trama queda fijada a un valor de 100 octetos.

A nivel físico, a los 127 octetos de la trama MAC, habría que sumar el preámbulo de sincronización (5 octetos), así como la cabecera física (1 octeto), de forma que el tamaño total a nivel físico es de 133 octetos.

- Sobrecarga del protocolo *Digimesh*: Como se ha indicado anteriormente, el tamaño máximo de carga útil ofrecido por el estándar 802.15.4 a nivel de aplicación es de 100 octetos. De estos 100 octetos, 27 se corresponden con la cabecera *Digimesh* y la suma de verificación, mientras en los 73 octetos restantes se almacena información de la carga útil del paquete. En el apartado anterior se indicaron los diferentes tipos de paquetes que se envían en el proceso de reprogramación. Atendiendo a los paquetes en los que se envía el nuevo código a cargar, éstos tenían una sobrecarga de 13 octetos, incluyendo el ID de cliente (1), ID de la carga útil (1), opciones (1), clave de reprogramación (8) y número de fragmento (2). Esto se traduce en que el espacio reservado para incluir datos del código enviado es de 60 octetos. Los campos ID cliente y opciones no se especificaron en la trama de datos de reprogramación (figura 4.5), puesto que no eran relevantes para el protocolo de reprogramación.
- Potencia de transmisión: El módulo XBee PRO presenta una potencia de transmisión máxima de 10 dBM (10 mW) y una sensibilidad de -100 dBm, presentando unos alcances teóricos máximos de 750m en exteriores (con línea de vista) y 60 m en interiores/entornos urbanos.
- Encriptación de la información: En aras de preservar la información transmitida entre los nodos de la red, las comunicaciones se cifran utilizando el estándar de seguridad AES (*Advanced Encryption Standard*) [AES] con claves de 128 bits de tamaño (AES128). Lógicamente, el uso de encriptación se traduce en una reducción de la tasa de datos, debido a la sobrecarga introducida por la misma. Es importante indicar que el cifrado no añade sobrecarga en términos de carga útil (se mantiene a 60 octetos) puesto que las cabeceras de seguridad se incluyen dentro de la cabecera *Digimesh*; pero sí implica una sobrecarga en tiempo de procesado de la trama, tanto en transmisión como en recepción, asociado al cifrado y descifrado de la trama, respectivamente.
- Parámetros propios de *Digimesh*: A continuación se muestra el valor de ciertos parámetros propio del protocolo *Digimesh*, adaptables en función de la red desplegada y que permiten estimar de manera teórica los tiempos de transmisión en función de los mismos:
  - Ranuras de retraso de la red (*Network Delay Slots, NN*): Fijado a un valor de 3 y que indica el máximo número aleatorio de slots de retraso, antes de reenviar un paquete de red (un slot de retraso de red aproximadamente equivale a 66 ms).
  - Saltos de la red (*Network Hops, NH*): Fijado a un valor de 7 y que fija el número máximo de saltos esperados en una ruta de la red. Este número no limita el número de saltos permitidos, pero se usa para estimar los tiempos de espera para recibir los reconocimientos a

nivel de red. Aunque en principio, los diferentes clusters se desplegaron intentando acotar el número máximo de saltos a 4, la opacidad del protocolo Digimesh (no se puede conocer el número de saltos asociado a una transmisión), así como la posible caída de nodos intermedios, justifica la elección de un valor superior para intentar asegurar la correcta transmisión/recepción de los paquetes para todos los nodos de un cluster.

- Retransmisiones a nivel MAC (*Mac Retries, RR*): Fijado a un valor de 3, especifica el número de reintentos que pueden ser enviados para un determinado paquete unicast, produciéndose un reintento cuando la confirmación (a nivel MAC) requerida por un paquete unicast supera el tiempo de espera máximo. Este es el número de retransmisiones a nivel MAC que se realiza en el protocolo de reprogramación en funcionamiento *unicast*. Si este parámetro se fija a 0, el paquete de confirmación no es requerido ni esperado y no habrá retransmisiones.
- Reintentos a nivel de red mallada (*Mesh Network Retries, MR*): Fijado a un valor de 1, especifica el número máximo de intentos de envío de un paquete a nivel de red. Si se fija a un valor diferente a 0, los paquetes enviados solicitarán un reconocimiento a nivel de red, y pueden ser reenviados NR+1 veces si los reconocimientos no han sido recibidos.
- Transmisiones múltiples (*Multiple Transmissions, MT*): Fijado a un valor de 3, especifica el número de veces que se retransmite un mensaje broadcast en la red. De esta forma, cada uno de los nodos retransmite MT+1 veces los paquetes broadcast que recibe, con el fin de intentar asegurar la correcta recepción del paquete por parte de todos los nodos que componen la red.
- Tiempo medio a un salto en comunicación *unicast* (unicastOneHopTime): Para un valor de RR=3, el valor es de 63 ms, indicando la cantidad de tiempo que conlleva una transmisión unicast entre dos nodos adyacentes.

- Tamaño de fichero: Se utilizan distintos tamaños de fichero (*Tam_fich*) oscilando entre 10 KB (1KB = 1Kocteto) y 120 KB (tamaño prácticamente máximo al tener la memoria *flash* donde se ejecuta una capacidad de 128KB) en intervalos de 10 KB. El tamaño del fichero binario depende principalmente de las librerías utilizadas, de si la interfaz de experimentación se encuentra activada o de las diferentes placas de sensores que se utilicen en función del tipo de nodo. Considerando el valor calculado anteriormente de la máxima carga útil ($L_{útil}$) de un paquete (60 octetos), se muestra en la tabla 4.1, el número de paquetes (N_{paq})a enviar en función del tamaño de la imagen de código, mediante la siguiente expresión:

$$N_{paqs} = \left\lceil \frac{1024 \cdot Tam\_fich(Koctetos)}{L_{útil}} \right\rceil \quad (4.1)$$

Tabla 4.1. Número de paquetes asociado a cada tamaño de código

Tamaño (KB)	10	20	30	40	50	60	70	80	90	100	110	120
Nº Paquetes	171	342	513	683	854	1025	1195	1366	1537	1707	1878	2049

- Establecimiento de la ruta: Como ya se ha comentado con anterioridad, *Digimesh* es un protocolo de enrutamiento propietario, lo que se traduce en que no ofrece información sobre las rutas obtenidas ni respecto al número de saltos entre los nodos. Este último punto, sin embargo,

puede ser inferido, modificando el parámetro *broadcast radius*, que limita el número de saltos que puede alcanzar una comunicación de tipo *broadcast* a nivel MAC. Incrementando gradualmente el valor de este parámetro, se descubren aquellos nodos que se encuentran a un diferente número de saltos de la pasarela. La tabla 4.2 muestra la distancia en número de saltos en los escenarios de interiores y exteriores.

Tabla 4.2. Distancia en saltos de los nodos en los escenarios de interiores y exteriores

Escenario	Saltos	Nodos
Interior (CAMPUS)	1	10000, 10001, 10002, 10003, 10004, 10005, 10007, 10009, 10010, 10011, 10012
	2	10006, 10008, 10013, 10015
Exterior (M06)	1	221, 222, 223, 225, 226, 229, 230, 2505
	2	215, 216, 217, 218, 228, 231, 434, 2508, 2510
	3	208, 209, 210, 214, 227, 430, 431, 432, 2509

Se debe tener en consideración que varios nodos pueden presentar comportamientos inestables en términos de distancia en número de saltos, pudiendo oscilar entre 1 y 2 saltos o 2 y 3 saltos, dependiendo de las condiciones de red. Por otro lado, es importante resaltar que, como este método de cálculo del número de saltos se realiza a través de una comunicación *broadcast*, el número de saltos podría no coincidir con aquel calculado por el protocolo *Digimesh*, el cuál puede establecer la ruta atendiendo a otro criterio diferente. Se utilizará esta tabla por tanto de manera orientativa, y en aras de intentar caracterizar de una manera más específica el escenario sobre el que se realizarán las medidas correspondientes.

Una vez definidas las características de la red, y para caracterizar el funcionamiento del protocolo desarrollado, se llevarán a cabo diferentes medidas de los valores de latencia, caudal eficaz y tasa de paquetes perdidos para los tamaños de fichero previamente indicados, comparando los modos de operación *unicast*, *multicast* y *broadcast,* así como la dependencia con la distancia en número de saltos de los nodos reprogramados.

4.5.1 Medidas *unicast*

En la figura 4.11, se muestran los valores medios de las latencias de las reprogramaciones a uno, dos y tres saltos en el despliegue de exteriores, así como a uno y dos saltos en el de interiores (no hay nodos a tres saltos en interiores), calculados como el valor medio de todas las reprogramaciones en todos los nodos a un salto, dos saltos y tres saltos según la tabla 4.2. Como se puede observar el comportamiento es lineal en ambos escenarios y para todas las distancias en número de saltos. Sin embargo,, se observa que las medidas asociadas al despliegue en interiores son ligeramente superiores a las de exteriores. Por otro lado la inestabilidad de los enlaces (traducida en la distancia entre nodos sobre todo de 2 a 3 saltos), se traduce en unos valores muy similares en las distancias a 2 y 3 saltos en exteriores, siempre superiores para los nodos situados a 3 saltos de distancia.

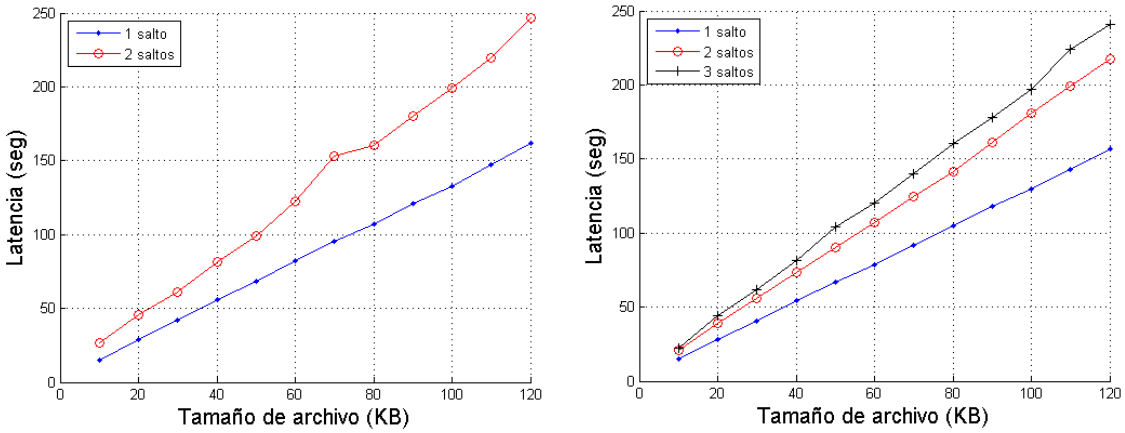

Figura 4.11. Latencia del protocolo en interiores (izquierda) y exteriores (derecha)

Teniendo en cuenta los valores de la tabla 4.1, respecto al tamaño de los ficheros, así como al número medio de paquetes (N_{paq}), se obtienen los valores de tiempo medio entre paquetes (t_{paq}), así como de caudal eficaz (C_{eficaz}), mediante las siguientes expresiones:

$$t_{paq} = \frac{Latencia}{N_{paqs}} \tag{4.2}$$

$$C_{eficaz} = \frac{8 \cdot N_{paqs} \cdot L_{útil}}{Latencia} \tag{4.3},$$

donde t_{paq} representa el tiempo medio entre paquetes, la *latencia* es el tiempo medio de reprogramación, N_{paq} es el número de paquetes asociado a los diferentes tamaños de fichero (tabla 4.1), C_{eficaz} es la tasa de transmisión de datos a nivel de aplicación, $L_{útil}$ es la máxima carga útil de datos de reprogramación (tamaño máximo de fragmento fijado a 60 octetos) y, finalmente, 8 es el factor multiplicativo para convertir octetos en bits, para expresar el caudal eficaz en bps (bits por segundo). La siguiente figura muestra los resultados asociados para uno, dos y tres saltos en las medidas realizadas en exteriores:

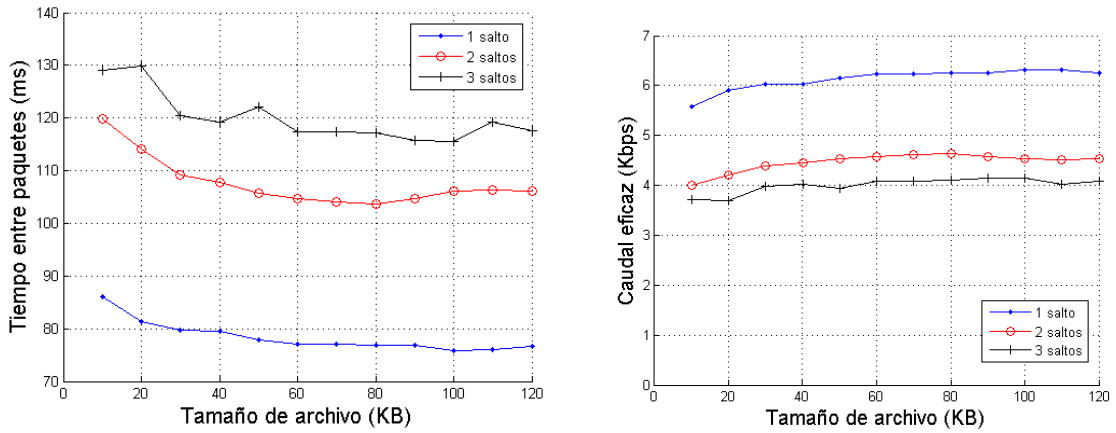

Figura 4.12. Tiempo entre paquetes (izquierda) y caudal eficaz (derecha) para el escenario de exteriores

Como se puede observar en la figura 4.12, tanto el tiempo entre paquetes como el caudal eficaz presentan un comportamiento prácticamente constante, siendo los resultados ligeramente peores para bajos tamaño de código, debido a que la sobrecarga asociada a la fase de negociación inicial y

final del protocolo tiene una mayor influencia al enviarse un número de paquetes inferior. Se observa que el caudal máximo a un salto se estabiliza en torno a los 6.5Kbps, lo que supone un 17% de la tasa de transmisión a nivel físico (38.4 Kbps). Respecto al tiempo medio entre paquetes a 1 salto, se estabiliza en torno a los 75ms, un valor muy similar al valor teórico del *unicastOneHopTime* (63ms). Por otro lado, respecto a los valores a dos saltos, tanto en tiempo medio entre paquetes como caudal eficaz, suponen un 50% de reducción respecto a los de un salto. Como se comentó con las medidas de latencia, los valores a tres saltos son bastante similares a los de dos saltos.

Es importante indicar que, en referencia, a la tasa de paquetes perdidos y número de retransmisiones pedidas por los nodos, éstas tienen un valor muy bajo (red considerablemente mallada) nunca superior a 10, y bastante heterogéneo ya que no mantiene un comportamiento relacionado con los tamaños de paquete que se envían (depende del estado del canal de comunicaciones), con lo que no se puede inferir un comportamiento específico asociado a la tasa de pérdida de paquetes. Es importante indicar que, aunque la tasa de paquetes perdidos (retransmisiones pedidas por los nodos destino) es baja, el protocolo es lo suficientemente robusto para que los nodos pidan la retransmisión de los fragmentos perdidos, así como para que gestionen convenientemente aquellos recibidos fuera de orden.

Por otro lado, también se ha estudiado el efecto del acceso a la memoria SD externa para el almacenamiento de los códigos enviados, cargándose 100 imágenes de código simultáneamente en la memoria SD y observando que el tiempo de reprogramación no variaba en función de la posición donde se almacenaba la imagen de código dentro de la memoria SD.

Por último, y como se comentó anteriormente, la velocidad del puerto serie en las comunicaciones de la pasarela con el módulo *Digi* puede aumentarse de los 38.4 Kbps hasta los 250 Kbps. A continuación, se muestra una comparativa entre ambas medidas:

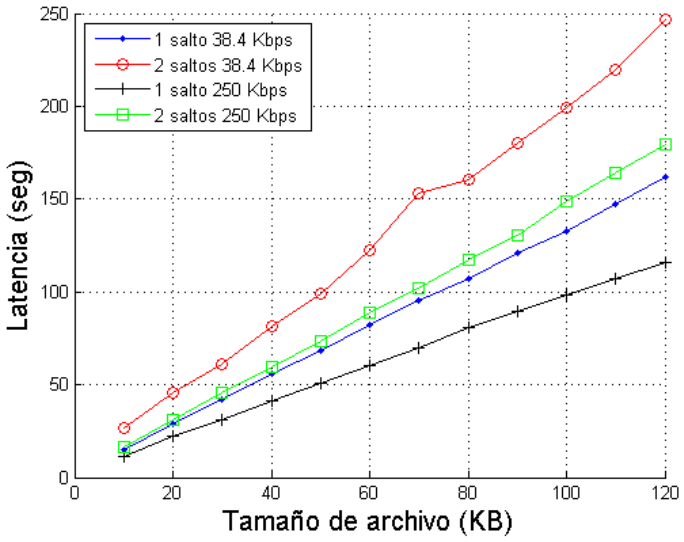

Figura 4.13. Comparativa de valores de latencia a 38.4 Kbps y 250 Kbps

De la figura 4.13, se desprende que el valor de las latencias a 38.4 Kbps es superior al de las latencias a 250Kbps, presentando una mejora de alrededor de entre 3 y 50 segundos en función del tamaño de fichero, aumentando la diferencia temporal con el número de paquetes transmitidos. Además, se puede constatar que los valores a dos saltos a 250 Kbps son muy similares a los de un salto a 38.4

Kbps. Aunque la mejoría a 250K bps es bastante grande, el problema es que sólo aquellos módulos *Digimesh* conectados a un puerto USB de la pasarela pueden trabajar a esta velocidad. En este caso, en la pasarela del *cluster* 6 el módulo *Digimesh* está conectado a un puerto serie y, por lo tanto, se ha limitado a 38.4Kbps, comparando las medidas entre interiores y exteriores a 38.4 Kbps. Esta comparativa entre latencias a 250 Kbps y 38.4 Kbps ha sido realizada en el laboratorio donde el módulo *Digimesh* se encuentra conectado a un puerto USB.

4.5.2 Medidas *multicast/broadcast*

Una vez analizado el comportamiento a nivel *unicast*, se realiza la comparativa de ambos escenarios con reprogramaciones a nivel *multicast/broadcast*, como se muestran a continuación:

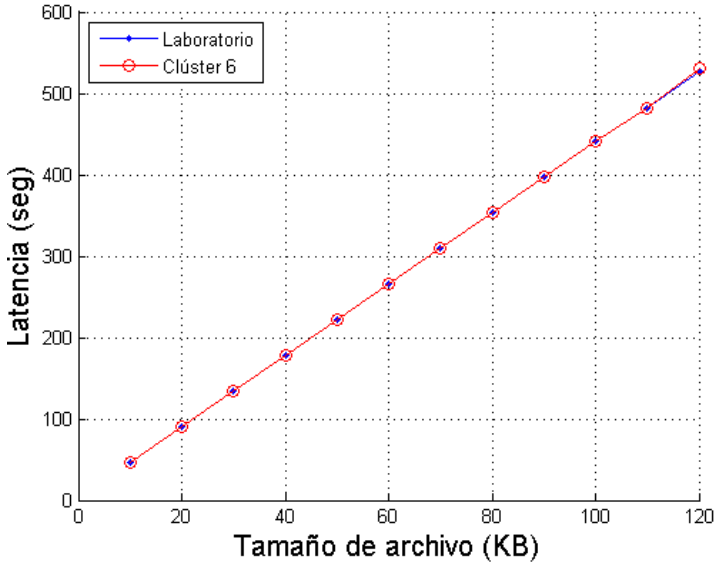

Figura 4.14. Valores de latencia multicast en interiores y exteriores

En la figura 4.14, se muestra que los valores en *multicast* para ambos son prácticamente idénticos y, al igual que ocurría en el caso *unicast*, el *multicast* también presenta un comportamiento lineal, con un tiempo medio de reprogramación constante e independiente del número de saltos a los que se encuentran los nodos que se reprograman. Este valor es aproximadamente tres veces superior al de reprogramación *unicast* a un salto y el doble que el de reprogramación *unicast* a 2 o 3 saltos. Es importante indicar que este valor constante es consecuencia del temporizador fijo que se incluye en el envío de los paquetes de un valor de 250 ms (casi el doble del tiempo medio entre paquetes para 3 saltos). Antes de fijar este valor se probaron valores inferiores, como por ejemplo 75 ms (tiempo medio entre paquetes para comunicaciones a 1 salto), que genera un gran número de paquetes erróneos/retransmisiones para tamaños de fichero inferiores a 50 KBs, fallando completamente el proceso para tamaños de fichero superiores; o un temporizador de 150 ms (ligeramente superior al tiempo medio entre paquetes para 3 saltos), que para comunicaciones *multicast* con nodos alejados (3 saltos) y en condiciones de baja calidad del canal, se comporta de manera bastante inestable produciendo un considerable número de retransmisiones (con el consiguiente aumento del tiempo de reprogramación), para tamaños de fichero elevados.

A continuación, se muestran los datos asociados al tiempo medio entre paquetes, así como el valor del caudal eficaz.

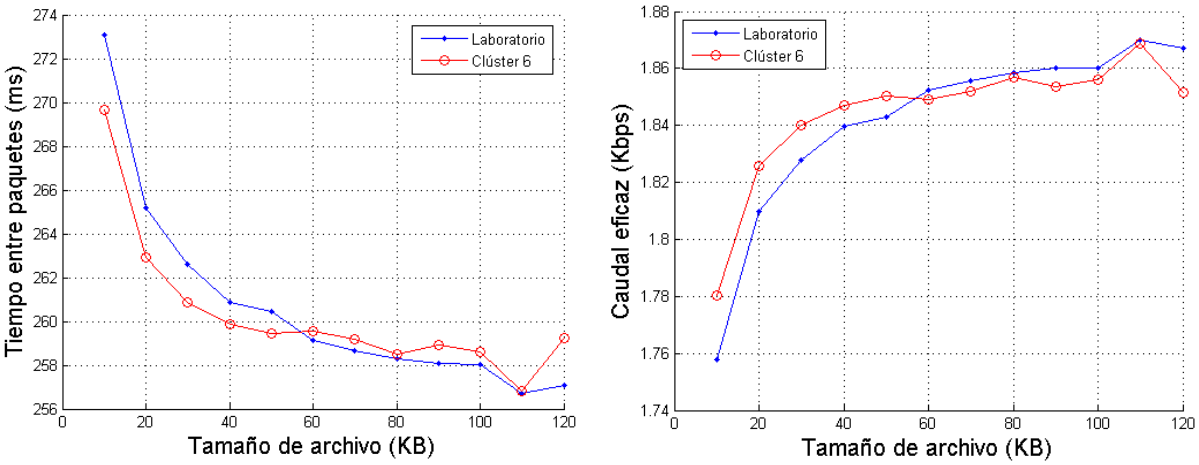

Figura 4.15. Valores de tiempo entre paquetes y caudal eficaz multicast en interiores y exteriores

Como se puede observar en la figura 4.15, el valor del tiempo medio entre paquetes se encuentra alrededor de los 260 ms, ligeramente superior a los 250 ms fijados en la fuente, ya que la presencia de alguna retransmisión puede provocar el aumento de este tiempo. Respecto al caudal eficaz, el valor se encuentra en torno a 1.85 Kbps, una tercera parte del valor *unicast*.

A continuación se muestran las medidas para la ejecución *broadcast* en ambos escenarios.

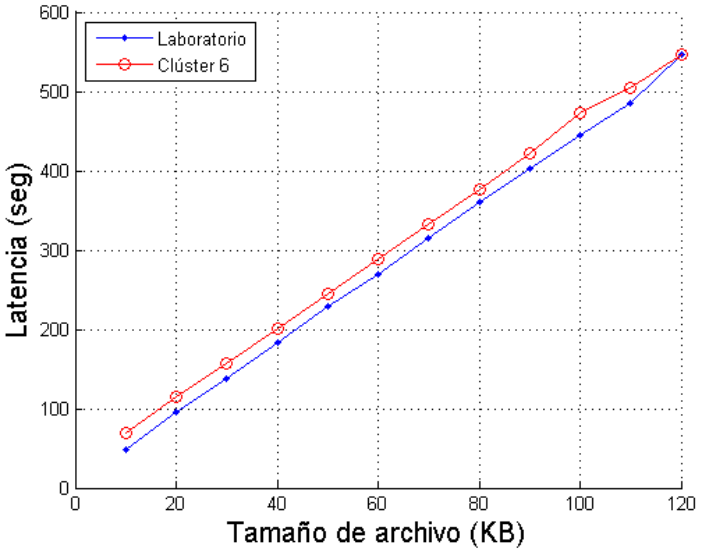

Figura 4.16. Valores de latencia broadcast en interiores y exteriores

En las figura 4.16 y figura 4.17, como era de esperar, se observa un comportamiento similar en *broadcast* y en *multicast*, ya que este modo de operación es realmente un *broadcast* que sólo escuchan el conjunto de nodos correspondientes. En este sentido, se observa una pequeña diferencia entre ambos escenarios, siendo un poco peor el de exteriores, debido a que está compuesto por un mayor número de nodos (26 frente a 15) y a una distancia mayor en número de saltos (algunos nodos se encuentran a 3 saltos de distancia). Al igual que para el caso *unicas*t, la tasa de paquetes perdidos no es especialmente representativa ni en número ni relacionada con el tamaño de código. En condiciones de red más agresivas (red menos mallada, condiciones meteorológicas adversas), sí se podría observar un mayor número de retransmisiones con el consiguiente empeoramiento del rendimiento del protocolo en *unicast*, *multicast* y *broadcast*.

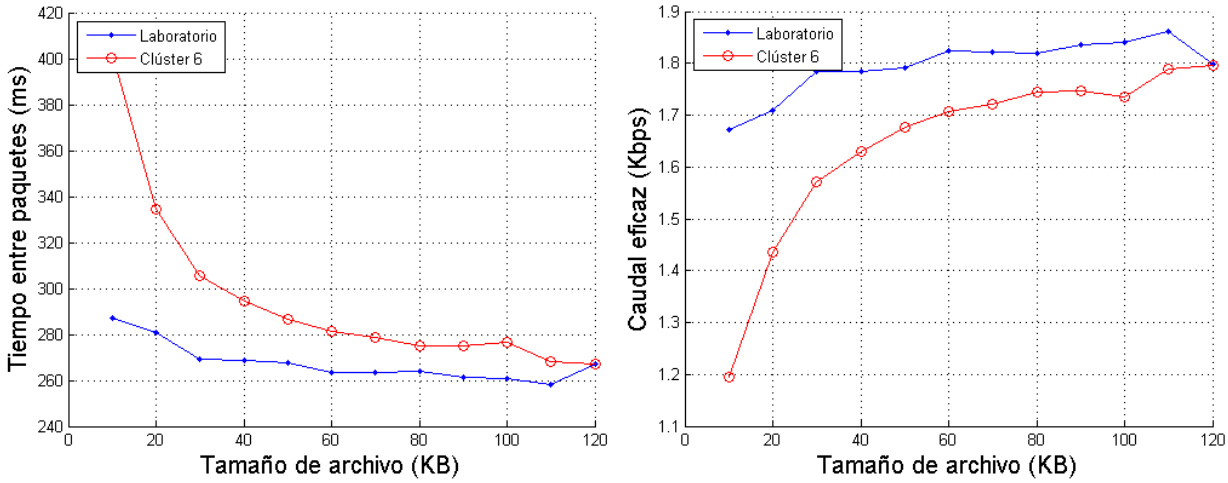

Figura 4.17. Valores de tiempo entre paquetes (izquierda) y caudal eficaz (derecha) broadcast en interiores y exteriores

Por último, al igual que se indicó en el caso un*icast,* en la figura 4.18 se realiza una comparativa entre los resultados obtenidos con una tasa del puerto serie de 250 Kbps en lugar de los 38.4 Kbps con los que se realizaron las medidas anteriores. En este caso, se observan que las medidas *multicast* son muy similares a las *broadcast*, independientemente del tamaño de paquete. Respecto a la diferencia entre los valores a 38.4Kbps y 250Kbps, oscila entre valores similares a los de *unicast*, con una diferencia máxima de alrededor de 50 segundos para el tamaño de fichero más alto.

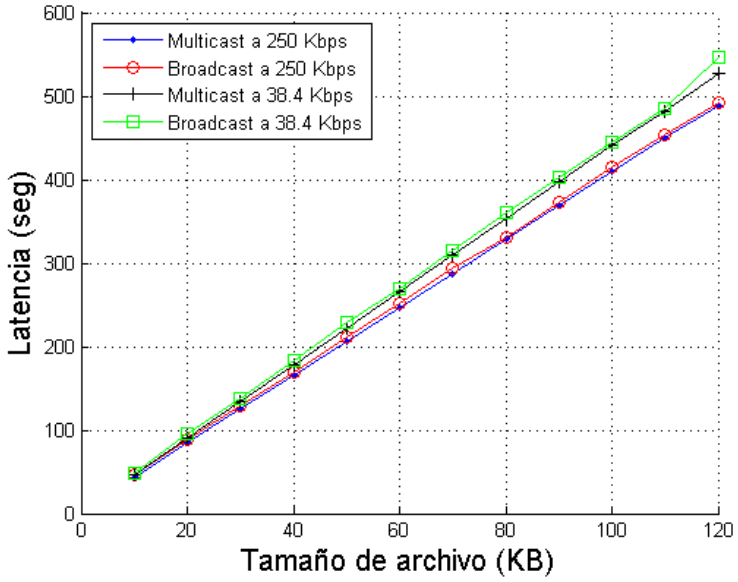

Figura 4.18. Valores de tiempo entre paquetes (izquierda) y caudal eficaz (derecha) broadcast en interiores y exteriores

Teniendo en cuenta las medidas, tanto multicast como broadcast, éstas tan sólo serán válidas cuando el número de nodos a reprogramar sea superior a 3. En caso contrario, la realización de procesos *unicast* sucesivos sería más eficiente.

4.6 CONCLUSIONES

Se ha desarrollado un protocolo de reprogramación remota que permite cargar tantas imágenes de código como sean necesarias sobre los dispositivos IoT despegados, permitiendo reconfigurar su comportamiento, tanto a nivel de experimentación como de servicio. Este protocolo ha sido diseñado atendiendo a las características específicas del despliegue y la arquitectura descritas en el Capítulo 3, permitiendo a las pasarelas y a los nodos que no sean objetivo de la reprogramación, continuar enviando/recibiendo los datos asociados a la experimentación y a la provisión de servicio, de manera concurrente a la ejecución del proceso de reprogramación sobre otros nodos.

Finalmente, se ha validado el protocolo diseñado, mediante la ejecución de distintas medidas de diferentes parámetros (latencia, caudal eficaz) en modo *unicast*, *multicast* y *broadcast*, a diferentes distancias en número de saltos y sobre escenarios en exteriores e interiores.

5 EXPERIMENTACIÓN A NIVEL DE NODO: DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DE UN EXPERIMENTO RELATIVO AL DESCUBRIMIENTO DE NODOS VECINOS

El principal objetivo de este capítulo consiste en demostrar la capacidad de experimentación a nivel de nodo que ofrece la arquitectura diseñada e implementada en el Capítulo 3, sobre los dispositivos desplegados en la red. En este sentido, tanto el protocolo de configuración remota de la red desarrollado en el Capítulo 4, que permite cargar tantos experimentos como sean necesarios sobre los nodos desplegados, como la interfaz adicional 802.15.4 nativa destinada a las transmisiones/recepciones asociadas con los diferentes experimentos, se erigen como las capacidades principales para llevar a cabo este tipo de experimentación.

Como experimento específico, se desarrolla en este capítulo un protocolo de descubrimiento de vecinos, basado en la inundación (flooding) de la red mediante el envío de tramas de difusión (broadcast), que permite elaborar la tabla de vecinos a nivel 802.15.4 de cada uno de los nodos de la red.

5.1 CONDICIONES DE CONTORNO

Existen aproximadamente 1.000 dispositivos IoT que ofrecen, adicionalmente a la provisión de servicio, la capacidad de experimentación sobre ellos, siendo reprogramados con el código correspondiente del experimento que pretende ser ejecutado sobre los mismos.

Dentro de los casos de uso que permiten la doble vertiente experimentación-servicio, monitorización medioambiental estática y móvil, así como riego inteligente, se ha seleccionado un conjunto de nodos pertenecientes a los mismos (específicamente a la monitorización medioambiental fija) para ejecutar el correspondiente experimento sobre ellos. Así, al mismo tiempo que estos nodos miden diferentes parámetros ambientales o de riego (según corresponda), serán reprogramados con el correspondiente experimento (descubrimiento de vecinos), implementado en este capítulo.

Como se ha comentado en capítulos anteriores, todos los nodos que ofrecen la capacidad de realizar experimentación sobre ellos, se encuentran provistos de una interfaz 802.15.4 nativa para el envío de las tramas asociadas al experimento correspondiente [Galache12]. Adicionalmente a esta interfaz, los nodos están provistos del citado módulo radio *Digimesh*, para la transmisión de los datos asociados a la provisión de servicio y la gestión de la red. La siguiente figura muestra el funcionamiento de los nodos desplegados donde, tanto los repetidores como las pasarelas, presentan la doble interfaz radio *Digimesh* y 802.15.4 nativa.

La figura 5.1 es muy similar a la mostrada a la figura 3.5, pero suprimiendo los nodos finales (mostrando sólo los repetidores y las pasarelas), puesto que éstos, principalmente por limitaciones de batería, no incluyen la interfaz 802.15.4 nativa y no permiten la experimentación a nivel de nodo sobre ellos. Es importante resaltar que el comportamiento de la red es distinto desde el punto de vista de la experimentación (interfaz 802.15.4 nativa), que desde el de la provisión de servicio/gestión de red (interfaz *Digimesh*).

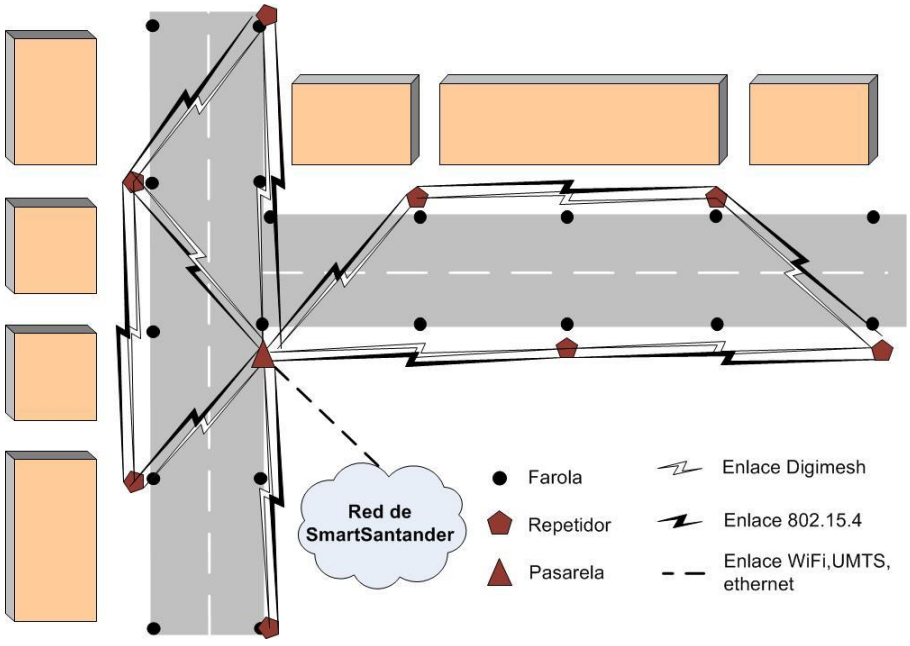

Figura 5.1. Plataforma dual Digimesh/802.15.4

Por un lado, desde el punto de vista de la interfaz *Digimesh*, debido al gran número de dispositivos desplegados, el procesado de los datos de servicio ofrecidos por los mismos, así como su gestión es muy complejo (en términos de número de saltos, cantidad de datos procesados, colisiones, interferencia), mediante el uso de una única pasarela. Para este propósito, varias pasarelas se han desplegado en la ciudad, formando diferentes *clusters* (asociados a una determinada área geográfica), que agrupan a un conjunto de nodos, los cuales envían los datos de provisión de servicio a su pasarela correspondiente, encargándose esta de gestionar este conjunto de nodos. De esta forma, se limita el número de saltos y la cantidad de información que se envía en cada uno de los *clusters* definidos. Con el objetivo de evitar problemas asociados a la colisión de paquetes y la interferencia, los *clusters* adyacentes se configuran con diferentes canales e identificadores de red (permiten formar grupos independientes dentro de un conjunto de nodos que se encuentran en el mismo canal).

Por otro lado, desde el punto de vista de la experimentación, todos los módulos 802.15.4 nativos presentan los mismos parámetros de configuración para toda la red, de forma que a nivel físico la red no se encuentra dividida en *clusters* (como ocurre en *Digimesh*), por lo que todos los nodos pueden comunicarse entre ellos. Sin embargo, esta comunicación queda restringida a los nodos vecinos, ya que la interfaz 802.15.4 nativa no implementa ningún protocolo de enrutamiento, de forma que los nodos sólo pueden interactuar con aquellos que se encuentren en su área de cobertura (nodos vecinos).

La realización de un determinado experimento a nivel de nodo sobre la plataforma desplegada se divide en las siguientes etapas:

- Los experimentadores deben autenticarse contra la plataforma de SmartSantander (a través del servidor SNAA ubicado en el servidor central), accediendo a la misma con los correspondientes permisos asociados a un experimentador, a saber: i) reserva de un determinado conjunto de nodos de la red, ii) reprogramación con la imagen de código correspondiente y iii) recepción de los resultados asociados al experimento realizado.
  En la figura 5.2, se muestra el proceso de autenticación frente a la plataforma por parte del experimentador, con objeto de acceder a la misma.
- El conjunto de nodos seleccionado para llevar a cabo un experimento se reserva (módulo RS del servidor central), indicando la duración del mismo, de forma que los nodos reservados no puedan ser ofrecidos a otros experimentadores durante el período de ejecución. El conjunto de nodos seleccionado no tiene porqué pertenecer a un solo *cluster*, sino que puede estar asociado a *clusters* diferentes.
  En la figura 5.3, se muestra en primer plano, el cuadro correspondiente para realizar la selección de aquellos nodos reservados que se desea reprogramar, ya que no todos los nodos reservados tienen que ser reprogramados con la misma imagen de código. Por su parte, en segundo plano se muestran los datos generados por los nodos reservados, tanto referentes al servicio como a resultados de un experimento (enviados a través de los denominados mensajes de *log*).

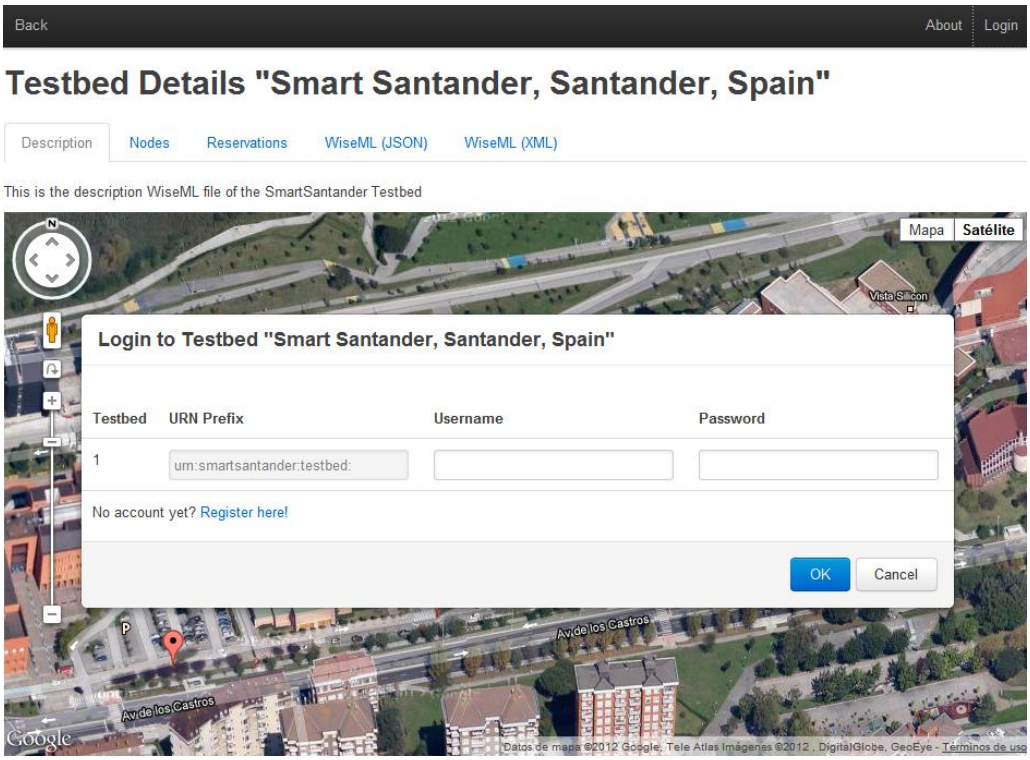

Figura 5.2. Autenticación contra la plataforma SmartSantander

- Los nodos seleccionados se reprograman con la correspondiente imagen de código del experimento a ser ejecutado sobre los mismos, utilizando el procedimiento de OTAP descrito en el Capítulo 4, y estableciendo la comunicación servidor central-pasarela-nodo a través de las correspondientes instancias del módulo iWSN, instalada en cada uno de ellos.

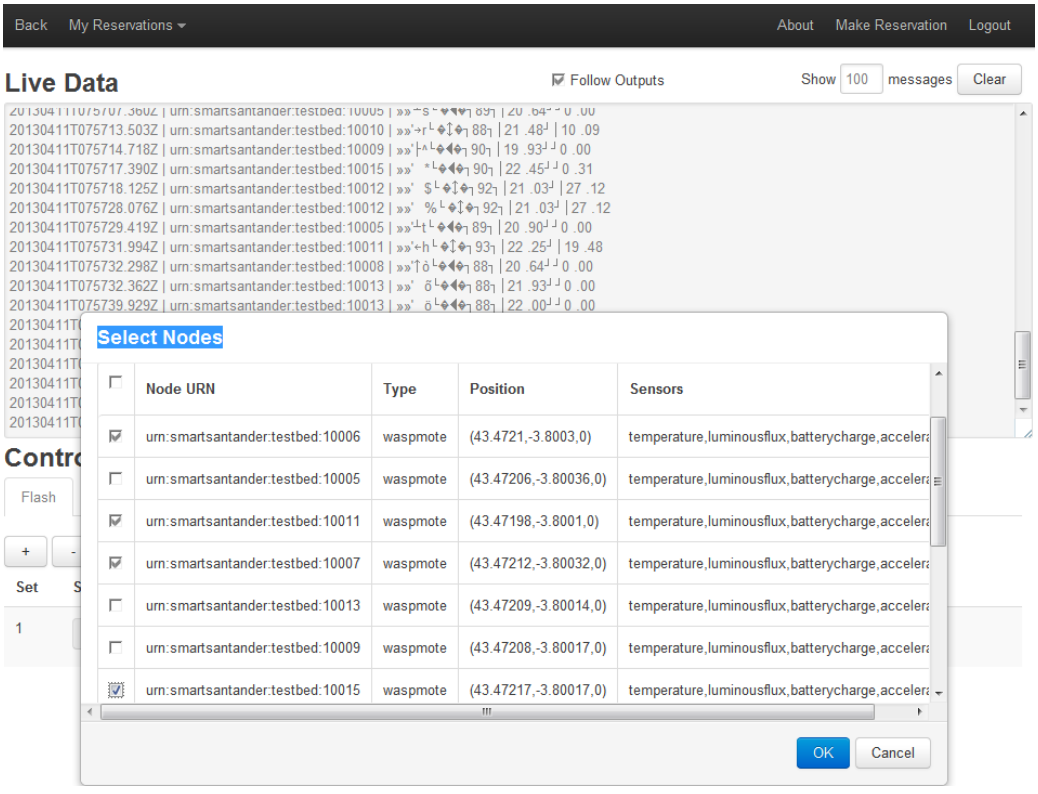

Figura 5.3. Selección y reserva de nodos

- Desde el punto de vista de la ejecución del experimento, se proveen al experimentador los métodos y funciones necesarias para el acceso, transmisión y recepción de paquetes a través de la interfaz 802.15.4. Sin embargo, el envío desde el nodo al experimentador de los resultados obtenidos en el experimento, se realiza a través de la interfaz Digimesh mediante los denominados mensajes de log. Este modo de funcionamiento se debe a que la interfaz 802.15.4 nativa no implementa ningún tipo de protocolo de enrutamiento y, por lo tanto, no permite la comunicación desde un nodo hacia nodos que no se encuentran dentro de la zona de cobertura del mismo. De esta forma, en aras de asegurar el correcto envío de los resultados desde cualquier nodo que intervenga en el experimento, se utiliza la interfaz Digimesh, enviando los nodos la información del experimento a su pasarela correspondiente. Para el envío de esta información, se habilita una función denominada *sendLog* que permite el envío de tantos mensajes de log como sean necesarios desde el nodo a su pasarela. Este tipo de mensaje se caracteriza por permitir el envío de texto libre incluyendo las medidas y resultados asociadas al experimento correspondiente.
- Las diferentes pasarelas que gestionen nodos que intervengan en el experimento, procesan los datos recibidos (mediante los mensajes de *log*) reenviándolos al servidor central, que lo distribuirá al experimentador correspondiente. Por tanto, las pasarelas no sólo se comportan como nodos que agregan tráfico asociado a los diferentes servicios provistos, y como nodos intermedios para la gestión de su *cluster* correspondiente, sino que también reciben y reenvían las medidas asociadas a un determinado experimento que se realice sobre uno o varios nodos dentro de su *cluster*.

Una vez detallado tanto el escenario de aplicación del experimento como el procedimiento específico para interactuar con la arquitectura desarrollada para llevar a cabo aquél, se detalla en el siguiente apartado la implementación práctica del mismo.

5.2 IMPLEMENTACIÓN PRÁCTICA

Con objeto de demostrar y explotar las capacidades de experimentación provistas por la plataforma desplegada, se ha implementado un protocolo de descubrimiento de vecinos basado en inundación. Este protocolo trasciende al concepto de vecino (nodo que se encuentra dentro de la zona de cobertura de otro nodo), extendiéndolo a nodos situados a varios saltos de distancia, con la intención de obtener un conocimiento más específico de la topología de la red en un momento determinado. La figura 5.4 muestra el diagrama de flujo que describe el funcionamiento del protocolo desarrollado, describiéndose a continuación los pasos en la realización del mismo:

- Una vez que un nodo comienza a ejecutar un proceso de descubrimiento de vecinos, éste realiza la inicialización de todas las variables que intervienen en aquél, vaciando las tablas de vecinos y de respuestas, inicializando los identificadores de descubrimiento y respuesta para evitar envío de paquetes duplicados, así como el contador de número de paquetes enviado que se incrementará cada vez que el nodo envía un paquete. Además, se activan las funcionalidades para la transmisión/recepción, tanto a nivel *Digimesh* como 802.15.4. A partir de este momento, el nodo está preparado para comenzar el proceso de descubrimiento.

- A continuación, la pasarela envía un paquete en modo *broadcast*, indicando la distancia máxima en número de saltos a la que se realiza el proceso de descubrimiento de vecinos, así como la correspondiente respuesta de los nodos. Para asegurar la recepción de este paquete por parte de todos los nodos del *cluster*, éste es enviado a través de la interfaz *Digimesh*, siendo procesado adecuadamente por los nodos que asignarán a los parámetros del máximo número de saltos de descubrimiento y de respuesta los valores recibidos (estos valores pueden ser diferentes como se indicará más adelante). Si en el experimento intervienen nodos pertenecientes a diferentes pasarelas, cada una de ellas se encarga de enviar el mencionado paquete a los nodos que pertenezcan a su *cluster*.

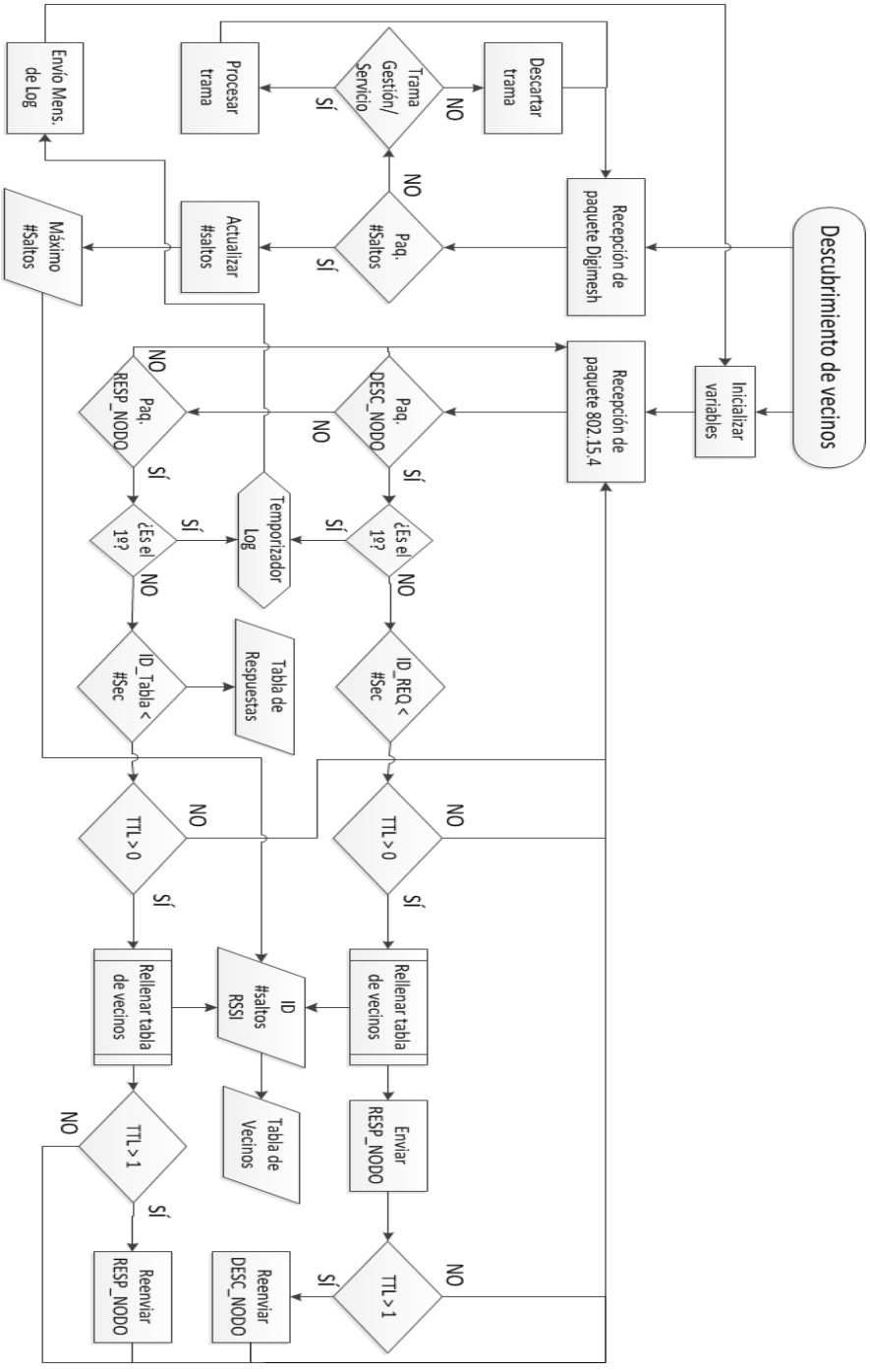

Figura 5.4. Diagrama de flujo del protocolo de descubrimiento de vecinos

- Una vez que los nodos han fijado el número de saltos máximo para el descubrimiento de vecinos, la pasarela procede al envío de un mensaje de descubrimiento de vecinos (paquete DESC_NODO), en modo *broadcast* a través de la interfaz 802.15.4, presentando la estructura siguiente:

Figura 5.5. Trama de descubrimiento de vecinos

Como se observa en la figura 5.5, el paquete de descubrimiento de vecinos incluye el correspondiente *identificador de paquete de descubrimiento*, junto con tres campos principales: el *TTL*, el *Num_Sec* y el *ID Nodo*. El *TTL* (*Time To Live*), desempeña una función similar a la que desarrolla el *TTL* implementado dentro de las redes IP, evitando que el paquete de descubrimiento de vecinos se propague de manera infinita en la red. Este parámetro se fijado al valor del máximo número de saltos, de forma que el paquete se propague hasta los nodos que se encuentren como máximo a esa distancia en número de saltos. Respecto al *Num_Sec*, éste representa el número de secuencia del paquete, el cual se fija a un valor inicial que se aumenta de manera secuencial permitiendo, de esta forma, diferenciar distintos procesos de descubrimiento, así como evitar el envío de paquetes asociados a procesos obsoletos. Por último, el ID Nodo especifica el identificador (se envía en formato ASCII ocupando entre 1 y 5 octetos dependiendo del número de cifras del identificador), asociado al nodo que comienza el proceso de descubrimiento (en este caso la pasarela). La pasarela, a diferencia de lo que ocurre a nivel *Digimesh* donde se erige como el nodo concentrador de un determinado *cluster*, a nivel 802.15.4 se comporta como cualquier otro nodo de la red de experimentación. No obstante, es importante reseñar que actualmente no se encuentra habilitada para los experimentadores la posibilidad de reservar y reprogramar las pasarelas, de forma que en este experimento se utilizará como iniciadora del protocolo de descubrimiento por su facilidad de acceso y configuración respecto a los nodos, pero no intervendrá en el resto del protocolo. De esta forma, la pasarela escuchará los paquetes recibidos a nivel 802.15.4, como si fuera otro nodo de la red que interviene en el proceso de descubrimiento, pero no los procesará ni reenviará. La pasarela enviará y recibirá los paquetes del experimento a través del correspondiente puerto de experimentación virtual (sobre el puerto físico 802.15.4), habilitado por el servidor de comunicaciones para este experimento concreto.

- Cuando un nodo recibe un mensaje de descubrimiento de vecinos, si éste es el primer mensaje (considerando tanto los mensajes de descubrimiento como los de respuesta) recibido por el nodo, éste inicia un temporizador (de 2 minutos), que delimita el período en el que el nodo se encuentra transmitiendo/recibiendo tramas asociadas al proceso de descubrimiento de vecinos (asociado a un determinado número de secuencia). A continuación, el nodo comprueba si el valor de *TTL* es mayor que 0 (paquete de DESC_NODO válido), introduciendo el valor del identificador del paquete de descubrimiento (en este caso ID de la pasarela) como un vecino dentro de su tabla de vecinos, incluyendo la distancia (en número de saltos) a la que se encuentra el mismo, delimitada por la siguiente fórmula:

$$d = Núm\_Saltos_{DESC} - TTL_{DESC} + 1 \quad (5.1)$$

donde d representa la distancia entre nodos, $Núm\_Saltos_{DESC}$ representa el número máximo de saltos del proceso de descubrimiento y TTL_{DESC} representa el valor del TTL del paquete de descubrimiento recibido.

Por ejemplo, los nodos que reciben el paquete de descubrimiento directamente de la pasarela (TTL = MAXnúm_saltos), rellenan su tabla de vecinos incluyendo el ID de la pasarela a 1 salto de distancia. Adicionalmente, para aquellos nodos que se encuentran a un salto de distancia, se incluirá dentro de la tabla de vecinos el valor de la relación señal a interferencia (RSSI), la cual permite determinar la calidad de los enlaces entre los nodos (se asume simetría en los enlaces). Para nodos a múltiples saltos de distancia no tiene sentido incluir este indicador, puesto que la RSSI se obtendría como una combinación de los enlaces correspondientes de la ruta seleccionada entre ambos nodos. Por su parte, si el número de secuencia del paquete no es igual (paquete ya procesado y reenviado) o menor (paquete obsoleto perteneciente a un proceso de descubrimiento anterior), al almacenado por el nodo como identificador de descubrimiento (inicializado a 0 al comienzo del proceso de descubrimiento), se actualiza el identificador de descubrimiento del nodo con el valor del número de secuencia extraído del paquete de descubrimiento, de forma que los mensajes de respuesta queden unívocamente asociados a un determinado mensaje de descubrimiento. A continuación, para indicar la recepción del paquete de descubrimiento, siempre que el nodo no haya respondido anteriormente a este número de secuencia de descubrimiento (identificador de descubrimiento), éste genera un paquete de respuesta de vecino (RESP_NODO), aumentando el contador de paquetes enviados por el nodo. La estructura del paquete de respuesta se muestra a continuación:

Figura 5.6. Trama de respuesta del nodo

Como se puede observar en la figura 5.6, la estructura del paquete de respuesta se compone de los mismos campos que la del paquete de descubrimiento, cambiando el identificador de paquete por aquel correspondiente al paquete de respuesta de vecinos. En este caso, el valor del *TTL* se fija al valor máximo de número de saltos del proceso de respuesta, de forma que se propague y sea recibido por todos los nodos situados a esta distancia de saltos. Finalmente, el nodo incluirá su identificador en el paquete de respuesta, de forma que, al igual que con el mensaje de descubrimiento, los nodos que reciban este paquete puedan completar su tabla de vecinos, determinando la distancia (en número de saltos) a la que se encuentra este nodo. Establecer el mismo valor para el número de saltos máximo de los paquetes de respuesta que para el del paquete de descubrimiento, se traduce en el aumento de las colisiones, así como la inclusión de una alta sobrecarga de tráfico dentro de la red. Para solventar esta problemática y para evitar posibles colisiones debidas a envíos simultáneos realizados por nodos adyacentes, se introduce un retardo aleatorio (tanto para el envío de paquetes de respuesta como para el reenvío de paquetes de descubrimiento). Por su parte, para evitar una sobrecarga de

tráfico, directamente dependiente del número de nodos intervinientes y del número saltos máximo fijado, se puede fijar el valor máximo de saltos de respuesta a un valor inferior al número máximo de saltos para descubrimiento (ambos valores incluidos en el paquete de configuración enviado al inicio por la pasarela).

Finalmente, el nodo comprueba si el valor del *TTL* es superior a 1 (el paquete debe seguir propagándose), reenviándose en este caso el paquete de DESC_NODO (aumentando el contador de paquetes enviados) con el mismo número de secuencia e identificador (el de la pasarela), pero disminuyendo en uno el valor del *TTL*, de forma que todos los nodos que reciban el paquete de descubrimiento puedan incluir en su tabla de vecinos el identificador de la pasarela (iniciadora del procedimiento de descubrimiento).

- Cuando se produce la recepción de un paquete de respuesta de vecino, si éste es el primer mensaje (considerando tanto los mensajes de descubrimiento como los de respuesta) recibido por el nodo, éste inicia un temporizador (de 2 minutos), que delimita el período en el que el nodo se encuentra transmitiendo/recibiendo tramas asociadas al proceso de descubrimiento de vecinos (asociado a un determinado número de secuencia). Tras ello, éste comprueba si el valor del *TTL* es mayor que 0 (paquete de RESP_NODO válido), introduciendo el valor del identificador del nodo que ha generado el paquete de respuesta en su tabla de vecinos, incluyendo la distancia en número de saltos que viene determinada por la Ec. (5.2), así como el valor de la RSSI (si el nodo se encuentra a un salto de distancia).

$$d = Núm\_Saltos_{RESP} - TTL_{RESP} + 1 \quad (5.2),$$

donde *d* representa la distancia entre nodos, $NúmSaltos_{RESP}$ representa el número máximo de saltos asignado a la respuesta y TTL_{RESP} representa el valor del TTL del paquete de respuesta recibido.
De esta forma, mediante la recepción de los diferentes paquetes de respuesta enviados por todos los nodos como respuesta al paquete de descubrimiento, se rellenan las tablas de vecinos del resto de nodos. Adicionalmente a la tabla de vecinos, se crea otra tabla que almacena los números de secuencia de las respuestas asociadas a cada nodo (en el caso del descubrimiento no es necesaria una tabla, ya que este paquete solo es enviado por el nodo iniciador). En esta tabla, se almacena el valor de los identificadores de los nodos de los paquetes de respuesta recibidos.

Una vez actualizadas la tabla de vecinos y la de respuestas, se compara si el valor del *TTL* del paquete de respuesta es superior a uno (debe seguir propagándose) y si el número de secuencia no es igual (paquete ya procesado y reenviado) o menor (paquete obsoleto perteneciente a un proceso de descubrimiento anterior), al almacenado dentro de la tabla de respuestas, y asociado al nodo que envío el RESP_NODO. Si se cumplen ambas condiciones, el nodo reenvía el paquete (aumentando el contador de paquetes enviados), disminuyendo el valor del *TTL* en una unidad y, tras ello, se actualiza con el número de secuencia recibido la tabla de respuestas de vecinos, incluyendo este número de secuencia asociado al identificador del vecino que generó el RESP_NODO. La elección del valor del máximo número de saltos de respuesta, acota el valor máximo del número de paquetes que se envía en la red. Este número máximo de paquetes (N_{max_paqs}), viene representado por la Ec. (5.3) cuando el número máximo de saltos del descubrimiento es el mismo que el

de la respuesta, mientras que la Ec. (5.4) muestra el valor del número máximo de paquetes para un número máximo de saltos para la respuesta fijado a 1.

$$N_{max\_paqs} = N_{NODOS} \cdot (N_{NODOS} + 1) + 1 \quad (5.3),$$

donde N_{max_paqs} es el número máximo de paquetes enviados en la red y N_{NODOS} es el número de nodos involucrados en el proceso de descubrimiento.

Como se puede observar el número máximo de paquetes queda determinado cuando el número de saltos máximo es tal que todos los nodos pueden descubrirse entre ellos. Esto implica que todo los nodos reenviarán tanto el correspondiente paquete de descubrimiento enviado por la pasarela así como todos los paquetes de respuesta (tantos paquetes como nodos intervienen en el proceso de descubrimiento). A este número hay que sumarle el paquete de descubrimiento enviado por la pasarela para comenzar el proceso de descubrimiento.

$$N_{\max\_paqs} = 2 \cdot N_{NODOS} + 1 \quad (5.4),$$

donde N_{max_paqS} es el número máximo de paquetes enviados en la red y N_{NODOS} es el número de nodos involucrados en el proceso de descubrimiento.

En este caso el número máximo de paquetes queda determinado, al contrario de lo que ocurría en el caso anterior, cuando el número de saltos máximo es tal que todos los nodos pueden descubrirse entre ellos, sino con el número máximo que asegure que el paquete de descubrimiento sea recibido por todos los nodos. Esto se debe a que, como los nodos sólo envían el paquete de respuesta a los nodos adyacentes ($Núm\_Saltos_{RESP}$ = 1), éste paquete no será reenviado por los nodos receptores, de forma que cada nodo reenviará el paquete de descubrimiento así como enviará su correspondiente paquete de respuesta. A este número hay que sumarle el paquete de descubrimiento enviado por la pasarela para comenzar el proceso de descubrimiento.

- Una vez que finaliza el período de descubrimiento (temporizador iniciado al recibir el primer paquete DESC_NODO o RESP_NODO ha expirado), los nodos envían su correspondiente tabla de vecinos mediante uno o varios (la tabla tiene un tamaño superior al tamaño máximo de paquete de la red) mensajes de *log*. La estructura de estos mensajes se observa en la figura siguiente:

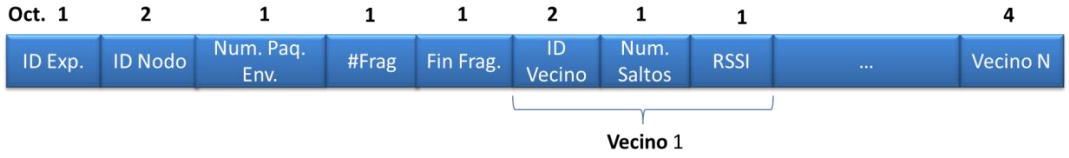

Figura 5.7. Mensaje de log de descubrimiento de un nodo

Como se puede observar en la figura, el paquete está constituido por el *identificador del experimento* (distinto identificador para diferenciar los resultados asociados a distintos experimentos), el *número de nodo* (en formato hexadecimal) que envía el paquete, el *número de paquetes enviados* (sobrecarga introducida en la red) por el nodo durante el proceso de descubrimiento, el *índice de fragmento* (pueden tener que enviarse varios paquetes en función del tamaño de la tabla) y, finalmente, el *indicador de final* que

especifica si se trata del último paquete o de uno intermedio (utilizado para procesar correctamente los paquetes de *log* recibidos). A continuación, se incluye el contenido de la tabla de vecinos, en la que para cada vecino se indica el identificador del mismo (2 octetos en formato hexadecimal), la distancia en número de saltos al nodo transmisor (del correspondiente mensaje de *log)* y el valor de la RSSI del enlace (este parámetro solo tiene un valor no nulo para los nodos a un salto de distancia). Una vez que se reciben estos paquetes en la pasarela o pasarelas (si intervienen en el experimento nodos pertenecientes a distintos *clusters*), éstos se procesan de la manera adecuada para constituir el mapa de vecinos de los nodos intervinientes.

Es importante reseñar que los nodos que intervienen en el experimento continúan enviando de manera simultánea los datos asociados a la provisión de servicio, así como procesando los paquetes referentes a la gestión de la red, de forma que el servicio no se interrumpe y el nodo es accesible y configurable en todo momento.

5.3 Caracterización de la implementación: Medidas y resultados

Para llevar a cabo la caracterización del protocolo de descubrimiento explicado, se ha realizado la instalación del código en un conjunto de nodos del despliegue en exteriores. En este caso, se seleccionan los nodos pertenecientes a dos *clusters* adyacentes (5 y 6), que se muestran en la figura 5.8, sobre el plano de la ciudad de Santander. La selección de dos *clusters* adyacentes permite demostrar la independencia a nivel físico entre las interfaces *Digimesh* y 802.15.4 nativa, así como las diferentes topologías de red asociadas a las mismas, en *clusters* para *Digimesh* y totalmente mallada para 802.15.4.

Figura 5.8. Vista general de los clusters 5 y 6

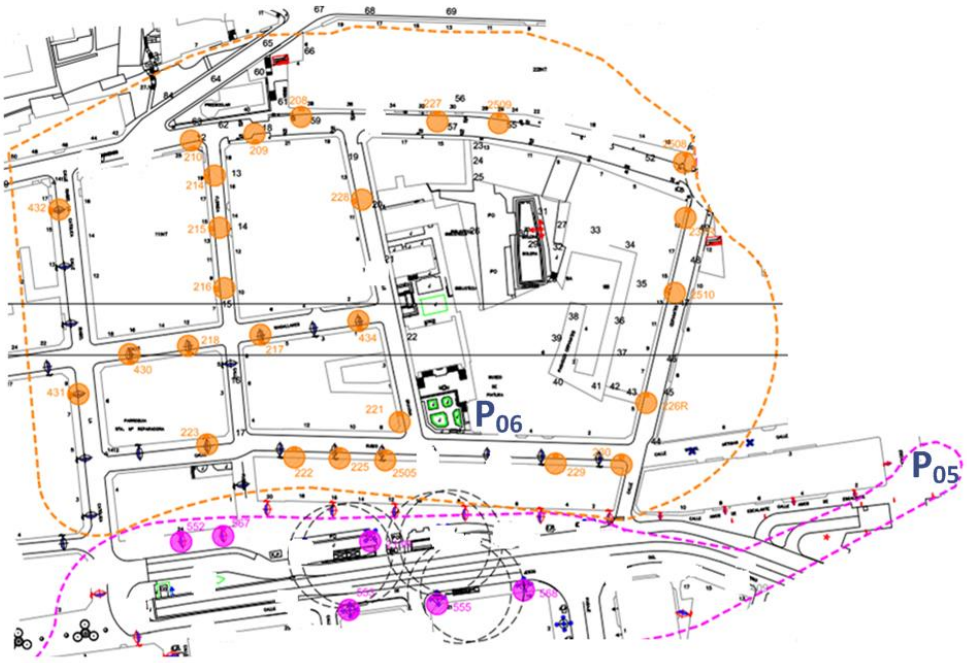

Figura 5.9. Detalle de los nodos de los clusters 5 y 6 que intervienen en las medida

Como se puede observar en la figura 5.9, se han seleccionado la mayoría de los nodos, 26, del *cluster* 6 (indicado en color naranja), así como algunos nodos, 6 del *cluster* 5 (indicado en color rosa), para demostrar la anteriormente citada interconexión a nivel 802.15.4, entre nodos pertenecientes a diferentes *clusters*. En este caso, el proceso de descubrimiento es lanzado por la pasarela del *cluster* 6 (P_{06}), la cual envía el correspondiente paquete de descubrimiento para iniciar el proceso. Se han llevado a cabo dos tipos de descubrimiento, uno de ellos en el que se asigna el mismo valor al máximo número de saltos para el descubrimiento y para la respuesta; y otro en el que el máximo número de saltos de la respuesta se fija a uno (sólo recibido por nodos adyacentes). Es importante resaltar que, como dentro del proceso de descubrimiento intervienen nodos pertenecientes a dos pasarelas diferentes, se han de combinar los resultados recibidos en ambas de ellas a través de los mensajes de *log* generados por los nodos intervinientes.

En aras de caracterizar el protocolo realizado, se realizarán diferentes medidas relacionadas con la topología de la red, la latencia del protocolo y la sobrecarga introducida por el mismo, variando el número máximo de saltos con el que se desarrolla el procedimiento de descubrimiento.

5.3.1 Topología de la red

Mediante la información de la distancia en número de saltos entre los nodos, así como el dato de la calidad de los enlaces entre nodos adyacentes (ambos parámetros incluidos en el paquete de *log* enviado por cada nodo), se puede caracterizar la topología de la red. A continuación se muestran tanto el mapa como la tabla de adyacencia de los 32 nodos y la pasarela que intervienen en el proceso de descubrimiento.

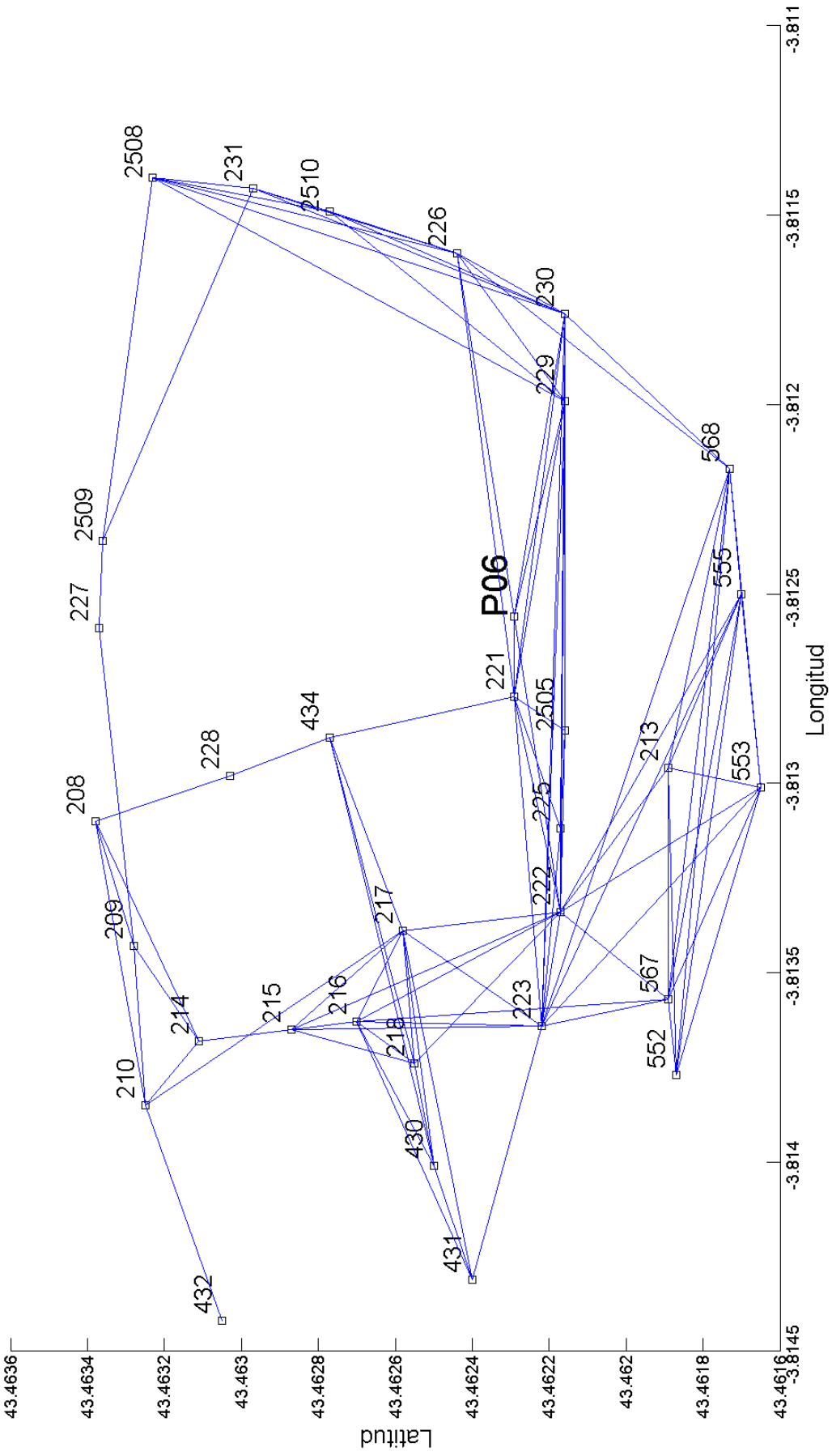

Figura 5.10. Mapa de adyacencia

Nodos	P06	208	209	210	213	214	215	216	217	218	221	222	223	225	226	227	228	229	230	231	430	431	432	434	552	553	555	567	568	2505	2508	2509	2510
P06	-	0	0	0	0	0	0	0	0	0	72	83	0	0	77	0	0	45	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0
208	0	-	52	60	0	68	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
209	0	52	-	50	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
210	0	60	50	-	0	67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0
213	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65	47	52	58	59	0	0	0	0	0
214	0	68	50	67	0	-	69	74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
215	0	0	0	0	0	69	-	52	59	67	0	80	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
216	0	0	0	0	0	74	52	-	40	63	0	69	72	0	0	0	0	0	0	0	0	0	0	0	0	0	74	0	0	0	0	0	0
217	0	0	0	67	0	0	59	40	-	57	0	69	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
218	0	0	0	0	0	0	67	63	57	-	0	72	0	0	0	0	0	0	0	0	45	0	0	73	0	0	0	0	0	0	0	0	0
221	72	0	0	0	0	0	0	0	0	0	-	66	72	72	76	0	0	64	68	0	0	0	70	0	0	0	0	0	57	0	0	0	0
222	83	0	0	77	0	80	69	69	72	66	-	57	52	0	0	0	0	83	79	0	0	0	0	0	77	76	76	0	63	0	0	0	0
223	0	0	0	0	0	0	80	72	76	0	72	57	-	63	0	0	0	75	82	0	0	75	0	0	0	63	71	69	78	0	0	0	0
225	0	0	0	0	0	0	0	0	0	0	72	52	63	-	0	0	0	0	77	0	0	0	0	0	0	0	0	0	65	0	0	0	0
226	77	0	0	0	0	0	0	0	0	0	76	0	0	0	-	0	0	68	60	62	0	0	0	0	0	0	0	0	79	0	65	0	52
227	0	0	72	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0
228	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	73	0	0	0	0	0	0	0	0	0
229	45	0	0	0	0	0	0	0	0	0	64	83	75	0	68	0	0	-	54	0	0	0	0	0	0	0	0	0	0	0	72	79	0	71
230	54	0	0	0	0	0	0	0	0	0	68	79	82	77	60	0	0	54	-	60	0	0	0	0	0	0	0	0	0	75	77	52	0	61
231	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62	0	0	0	60	-	0	0	0	0	0	0	0	0	0	0	0	51	73	50
430	0	0	0	0	0	0	0	68	57	45	0	0	0	0	0	0	0	0	0	0	-	57	0	68	0	0	0	0	0	0	0	0	0	0
431	0	0	0	0	0	0	0	73	66	0	0	0	75	0	0	0	0	0	0	57	-	0	0	0	0	0	0	0	0	0	0	0	0	0
432	0	0	0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0
434	0	0	0	0	0	0	0	0	60	73	70	0	0	0	0	0	0	0	0	0	68	0	0	-	0	0	0	0	0	0	0	0	0	0
552	0	0	0	65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	68	0	-	54	57	60	57	0	0	0	0	0
553	0	0	0	47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54	-	47	51	65	0	0	0	0	0
555	0	0	0	52	0	0	0	0	0	0	76	71	0	0	0	0	0	0	0	0	0	0	0	57	47	-	57	49	0	0	0	0	0	0
567	0	0	0	58	0	0	74	0	0	0	76	69	0	0	0	0	0	0	0	0	0	0	0	60	51	57	-	50	0	0	0	0	0	0
568	0	0	0	59	0	0	0	0	0	0	0	78	0	79	0	0	0	75	0	0	0	0	0	57	65	49	50	-	0	0	0	0	0	0
2505	0	0	0	0	0	0	0	0	0	57	63	0	65	0	0	0	0	72	77	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0
2508	0	0	0	0	0	0	0	0	0	0	0	0	0	65	0	0	0	79	52	51	0	0	0	0	0	0	0	0	0	-	75	57	0	0
2509	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	0	0	0	0	73	0	0	0	0	0	0	0	0	0	0	-	0	0	0
2510	0	0	0	0	0	0	0	0	0	0	0	0	0	52	0	0	0	71	61	50	0	0	0	0	0	0	0	0	0	0	0	0	0	-

Figura 5.11. Tabla de adyacencia

Tabla de vecinos pasarela P06

$NumSaltos_{DESC}$= 8

$NumSaltos_{RESP}$ = 1

Vecino	# Saltos	RSSI
208	5	0
209	4	0
210	5	0
213	4	0
214	5	0
215	3	0
216	2	0
217	2	0
218	2	0
221	1	72
222	1	83
223	2	0
225	2	0
226	1	77
227	4	0
228	3	0
229	1	45
230	1	54
231	2	0
430	3	0
431	3	0
432	6	0
434	2	0
552	4	0
553	3	0
555	4	0
567	4	0
568	5	0
2505	2	0
2508	2	0
2509	3	0
2510	2	0

Tabla de vecinos nodo 2509

$NumSaltos_{DESC}$ = 6

$NumSaltos_{RESP}$ = 6

Vecino	# Saltos	RSSI
231	1	73
221	3	0
2508	1	67
230	3	0
226	5	0
227	1	56
5006	4	0
228	5	0
2505	3	0
209	2	0
208	2	0
223	5	0
222	5	0
214	3	0
2510	4	0
217	6	0

Tabla de vecinos nodo 2509

$NumSaltos_{DESC}$ = 7

$NumSaltos_{RESP}$ = 1

Vecino	# Saltos	RSSI
2508	1	75
231	1	73
P06	3	0
227	1	55

Figura 5.12. Tablas de vecinos de la P06 (izda) y del nodo 2509 (dcha)

En la figura 5.11, se muestra la relación de adyacencia entre los nodos desplegados indicando la RSSI asociada a cada uno de ellos. Respecto a la figura 5.10, la gran cantidad de nodos intervinientes en el proceso de reprogramación dificulta la correcta visualización de todos los enlaces entre los nodos.

Adicionalmente a la matriz de adyacencia y a su representación gráfica, el protocolo desarrollado permite elaborar la tabla de vecinos de todos los nodos que intervienen en el proceso de descubrimiento. Por un lado, y como iniciadora del proceso de descubrimiento, la pasarela (P_{06}) siempre conforma su tabla de vecinos completa (utilizando la información de los mensajes de *log* enviados por los nodos), descubriendo nodos a un número de saltos máximo determinado por el número de saltos fijado para el descubrimiento. Por otro lado, sin embargo, desde el punto de vista del resto de nodos la conformación de su tabla de vecinos dependerá del número máximo de saltos establecido para la respuesta. Si éste es el mismo que el establecido para el descubrimiento, entonces cada nodo descubrirá nodos a la misma distancia en saltos que lo hace la pasarela; sin embargo para un número de saltos para la respuesta fijado a 1, cada nodo sólo descubrirá sus vecinos adyacentes. A continuación se muestra un ejemplo tanto de la tabla de la pasarela como de la de un vecino.

En la figura 5.12, se muestran las tablas de vecinos de la pasarela, así como del nodo 2509. La tabla de vecinos de la pasarela se obtiene mediante un proceso de descubrimiento a 8 saltos con un máximo de número de saltos de la respuesta fijado a 1. Respecto al nodo 2509, se muestra en la parte superior, una tabla de vecinos obtenida con número de saltos de descubrimiento y de respuesta fijados a 6, donde se observan todos los vecinos de este nodo a un máximo de 6 saltos de distancia. Sin embargo, en la tabla de la parte inferior, obtenida con un valor de número máximo de saltos de vecinos fijado a 7 y un valor de número máximo de saltos de respuesta fijado a 1, sólo aparecen los nodos a un salto de distancia y la pasarela (ésta siempre parece ya que se inserta en la tabla con la recepción del paquete de descubrimiento). Lógicamente, la tabla de la parte superior es más completa que la de la parte inferior (en la que sólo se descubren los nodos adyacentes). Como se puede observar, en ambas tablas, así como en la tabla de adyacencias mostrada en la

figura 5.11 (todas las tablas obtenidas como resultado de diferentes realizaciones del experimento), los nodos 227, 231 y 2508, aparecen como nodos a un salto de distancia del nodo 2509, presentando todos ellos unos valores de RSSI bastante similares.

5.3.2 Latencia del protocolo

Este parámetro se estima a partir de la diferencia de tiempo entre el envío del paquete de descubrimiento y la recepción del último paquete (ya sea de descubrimiento o de respuesta) en la pasarela, asociado a un proceso de descubrimiento específico (definido por el número de secuencia del paquete de descubrimiento), entendiendo este intervalo temporal como una extensión del tiempo de retardo de ida y vuelta de la red (*Round-Trip delay Time, RTT*). Como es la pasarela del *cluster* seis la que comienza el protocolo de descubrimiento, la latencia en este sentido se mide como la diferencia de tiempo entre el instante en que se envia el paquete de descubrimiento y el último mensaje de respuesta/descubrimiento recibido. Este valor temporal es claramente dependiente del máximo número de saltos asignado al descubrimiento y a la respuesta de los nodos.

A continuación, se muestran las medidas de latencia asociadas al proceso de descubrimiento variando el valor máximo de número de saltos de descubrimiento, tanto para el caso en que este valor es el mismo que el máximo número de saltos de respuesta, así como cuando este valor queda fijado a 1.

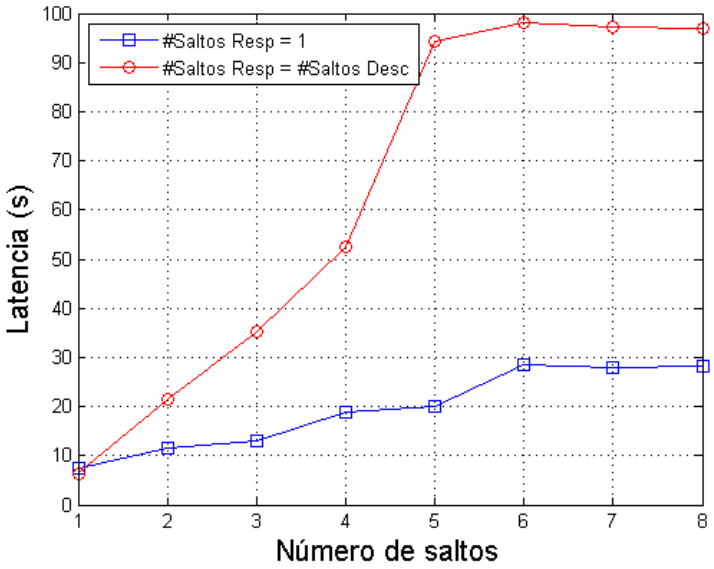

Figura 5.13. Latencia del proceso de descubrimiento (mismo valor del número de saltos)

Como se puede observar en la figura 5.13, cuando el número máximo de saltos de descubrimiento es igual al de respuesta, el valor de la latencia crece con el número de saltos estabilizándose a un valor de casi 98 segundos, cuando el número máximo de saltos se fija a un valor de seis. Esta medida es bastante coherente con el funcionamiento obtenido anteriormente en la topología, donde el nodo más alejado de la pasarela se encuentra a 6 saltos de distancia, de forma que a partir de ese número de saltos ningún nodo adicional reenviará ni un paquete de descubrimiento, ni uno de respuesta.

Sin embargo cuando el número máximo de saltos en la respuesta se fija a 1, la latencia también es creciente con el valor del número de saltos, estabilizándose también el valor cuando el número de saltos es 6. En este caso, el valor de latencia en el estado estable es de aproximadamente 30 segundos, muy inferior al tiempo definido anteriormente. Esto es debido a que, a diferencia de lo que ocurría en el caso anterior donde los paquetes de respuesta delimitaban la latencia del protocolo, en este caso como el número máximo de saltos de respuesta está acotado, será la recepción de paquetes de descubrimiento reenviados y escuchados por la pasarela, la que determine la latencia del protocolo.

5.3.3 Sobrecarga de la red

Este parámetro se asocia con la cantidad de paquetes que se envían dentro de la red durante el proceso de descubrimiento, obtenido como la suma del número de paquetes enviados por cada uno de los nodos (valor incluido dentro del paquete de *log*). Al igual que en los dos apartados anteriores, se distinguen los distintos comportamientos en función del máximo número de saltos del descubrimiento y de la respuesta.

A continuación, se muestran las medidas de sobrecarga de la red asociadas al proceso de descubrimiento variando el valor máximo de número de saltos de descubrimiento, tanto para el caso en que este valor es el mismo que el máximo número de saltos de respuesta, así como cuando este valor queda fijado a 1.

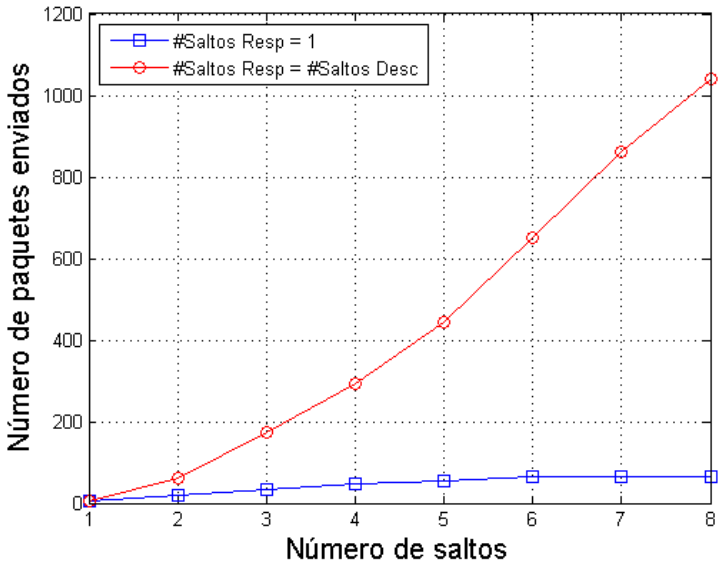

Figura 5.14. Sobrecarga del proceso de descubrimiento

Como se puede derivar de la figura 5.14, cuando el número máximo de saltos de descubrimiento es igual al de respuesta, el valor del número de paquetes enviados crece proporcionalmente con el número de saltos. En principio, esta tendencia se corresponde con el comportamiento esperado, puesto que el número de paquetes enviados crece con el número de saltos, tendiendo al valor máximo obtenido ($N_{MAXPAQS}$ = 1057) mediante la Ec. (5.3), asignando a N_{NODOS} el valor de 32, los nodos que intervienen en el proceso de descubrimiento. Esto se traduce en que, aunque el máximo número de saltos entre la pasarela y un nodo sea 6, eso no implica que el máximo número de saltos entre dos nodos sea superior (no recibirán los paquetes de respuesta mutuo y, por lo tanto, no los reenviarán), por lo que a medida que crece el número de saltos crecerá el número de paquetes enviados hasta un máximo de 1057 paquetes, que es el valor que prácticamente se obtiene con el número de saltos fijado a 8 (1039 paquetes).

A diferencia del caso anterior, cuando el número máximo de saltos en la respuesta se fija a 1, sí que se alcanzará (para un número de saltos superior a cinco), el valor máximo ($N_{MAXPAQS}$ = 65) expresado por la Ec. (5.4), asignando a N_{NODOS} el valor de 32, los nodos que intervienen en el proceso de descubrimiento. Esto se debe a que, en este caso, como los nodos no reenvían los paquetes de respuesta (*TTL*=1), que reciben de otros nodos, el número máximo de paquetes enviados en la red se producirá cuando todos los nodos hayan recibido el paquete de descubrimiento, que sucede para número de saltos fijado a 6 o mayor (como se demostró en el apartado de topología).

5.4 CONCLUSIONES

En este capítulo se ha demostrado la capacidad de experimentación a nivel de nodo que ofrece la arquitectura desarrollada, describiendo los pasos que debe realizar un experimentador para acceder a la plataforma, reservar el conjunto de nodos que precise, programar remotamente estos nodos con la imagen de código del experimento a realizar y recibir los resultados correspondientes derivados del citado experimento.

Para validar la mencionada capacidad de experimentación, se ha diseñado e implementado un protocolo de descubrimiento de vecinos que se ha cargado sobre un conjunto de nodos pertenecientes al despliegue en exteriores, caracterizando su correcto funcionamiento mediante medidas de latencia y sobrecarga, así como obteniendo las tablas de vecinos de los nodos intervinientes en el experimento.

6 SERVICIO: GESTIÓN ADAPTATIVA DE LA OCUPACIÓN DE LAS PLAZAS DE APARCAMIENTO

En los Capítulos 4 y 5 se han cubierto dos de los tres pilares, gestión de la red y experimentación, que cubre la arquitectura propuesta en el marco de este trabajo. Éste aborda el tercero de estos pilares, a saber, la provisión de servicio. Para ello se tomará como referencia uno de los servicios, asociado al caso de uso de gestión de aparcamiento en exteriores, incluyéndose un sucinto estado del arte de los servicios de este tipo existentes en la actualidad.

Con el objetivo de mejorar el rendimiento del servicio desplegado, se modificará para hacerlo más fiable así como fácilmente gestionable de manera remota, de forma que pueda adaptarse de manera dinámica a diferentes condiciones del entorno, así como a actualizaciones periódicas de software/firmware, comparándose el funcionamiento de los mismos.

6.1 INTRODUCCIÓN Y ESTADO DEL ARTE

6.1.1 Introducción

La creciente masificación de las ciudades deriva ineludiblemente en un aumento del parque de vehículos asociado a las mismas y, consecuentemente, en un incremento de movilidad urbana, liderado por desplazamientos de vehículos privados en las ciudades y la posterior necesidad de estacionamiento de los mismos. Esta situación conlleva la necesidad de realizar una gestión eficiente de plazas de aparcamiento, que la ciudad inteligente permite a través de la implementación de aplicaciones basadas en la recogida de información que emana del despliegue de sensores en la infraestructura.

Se han realizado estudios del tráfico en las grandes ciudades americanas confirmándose la influencia que tiene la problemática del aparcamiento en la congestión de tráfico, especialmente en los centros urbanos. En [Markoff08], se presentan los resultados de un estudio realizado en la ciudad de Nueva York en el que se concluye que hasta el 45% del tráfico en Manhattan era generado por coches en busca de una plaza de aparcamiento; estudios parecidos demuestran que, en Los Ángeles, a lo largo de un año un vehículo que está buscando una plaza de aparcamiento pueden consumir hasta 177.000 litros de gasolina produciendo 730 toneladas de dióxido de carbono. Estos datos, lógicamente, dependen de muchos factores (tamaño de ciudad, tipo de coche), pero ofrecen una visión clara de la problemática asociada a la gestión de plazas de aparcamiento, tanto a nivel de tráfico, como de consumo y medioambiental.

Actualmente, las necesidades de estacionamiento en las ciudades y la mejora de movilidad urbana, demandan la gestión eficiente de plazas de. El aparcamiento inteligente tiene como objetivo facilitar el proceso de búsqueda de plazas de aparcamiento, ofreciendo una solución integrada que dirija al conductor directamente a la plaza de aparcamiento más cercana a su destino final. La solución debe, por tanto, estar integrada con las tecnologías de la información y la comunicación que utilizan habitualmente los ciudadanos, tales como teléfonos inteligentes, tabletas o dispositivos de navegación personal. De esta manera, se pueden incluir los siguientes como principales detonantes para la implementación de este tipo de soluciones:

- Una reducción en el tiempo de búsqueda de aparcamiento es equivalente a una ganancia en dinero, principalmente para aquellas personas que dependen del vehículo para desarrollar su actividad profesional.
- Una búsqueda más eficiente de plaza de aparcamiento se traduce en una reducción de las emisiones de gases a la atmósfera, con el consiguiente impacto en la conservación del medio ambiente.
- Existe un creciente interés por parte de los ayuntamientos en la reducción del tráfico, mediante una búsqueda y una utilización más eficiente de las plazas de aparcamiento.
- Las tecnologías para implementar y ofrecer soluciones de aparcamiento inteligente están madurando y empiezan a producirse despliegues masivos asociados a ellas.

Las soluciones de aparcamiento inteligente son ya una realidad en el caso de los aparcamientos subterráneos, donde los conductores son guiados hasta las plazas libres mediante flechas y luces en

paneles luminosos, así como indicadores luminosos en cada plaza indicando el estado de ocupación de la misma. Sin embargo, estas soluciones se caracterizan por ser en su mayoría de tipo cableado, acotadas a entornos de interiores y, por tanto, su implementación no es aplicable para gestionar el aparcamiento en exteriores en las calles de una ciudad. La tecnología inalámbrica para poder dotar a las ciudades del servicio de aparcamiento inteligente está basada, principalmente, en un sensor (de diferentes tecnologías de detección) colocado dentro de una cápsula enterrada o adherida al asfalto en cada plaza de aparcamiento. El sistema de sensores distribuidos en las plazas de aparcamiento conforma una red mallada de comunicaciones inalámbricas que se conecta a través de uno o varios repetidores con una pasarela, que se encarga de procesar los datos recibidos y enviarlos al correspondiente centro de control/servidor central.

El principal reto para la implementación del servicio inteligente es la reducción del coste de implantación y mantenimiento hasta unos niveles que hagan viable su aplicación y la integración de tecnologías innovadoras con mecanismos económicos basados en reglas de mercado, que se han revelado más eficientes que los mecanismos convencionales de gestión del tráfico.

6.1.2 Estado del arte

El requerimiento principal de la monitorización de plazas de aparcamiento en superficie, en entornos ya urbanizados, reposa en la necesidad de que los elementos que se necesitan desplegar no obliguen a la realización de una instalación excesivamente invasiva. Ello se traduce en la necesidad de utilizar sistemas de comunicación inalámbricos, de bajo consumo y alimentados autónomamente mediante baterías. Ante estos requerimientos, las redes de sensores inalámbricos se convierten en una solución óptima para abordar la problemática descrita.

Desde el punto de vista de la detección dos son las tecnologías más habitualmente implementadas:

- Sensores magnéticos: El principio de funcionamiento de esta tecnología se basa en la perturbación que los elementos magnéticos, el vehículo en este caso concreto, generan en el campo magnético de la Tierra, asimilable a lo que se denomina “anomalía magnética”. El campo magnético en cada punto de la superficie terrestre presenta un valor característico, que se altera por la aparición en las proximidades de un cuerpo metálico, siendo esta alteración detectada por el sensor ferromagnético. Estos dispositivos suelen componerse por una bobina (hay soluciones dobles), que por inducción, percibe estas variaciones. La implantación de un sistema de este tipo implica dos pasos fundamentales:
  - Instalación de los sensores. Existen dos maneras de llevar acabo la instalación, a saber, encastrándolos en el pavimento o adhiriéndolos al mismo. En la primera de ellas se realiza un orificio en el suelo introduciendo el correspondiente sensor (con su cápsula protectora) en el pavimento, mientras que en el segundo se adhiere directamente el sensor a la calzada. Este segundo método presenta una instalación más rápida, pero con un mayor impacto visual. Por otro lado, desde la perspectiva de vandalismo, encastrar el dispositivo en el suelo (en alguna ocasión cubriéndolo completamente), presenta una solución más eficiente contra cualquier intento de acceso no autorizado al dispositivo.
  - Calibración inicial. Como el sistema de detección se basa en una alteración de una situación inicial característica de cada punto de la superficie terrestre, se precisa realizar una primera caracterización de las condiciones de contorno iniciales. Por lo tanto, con las plazas

desocupadas, se realiza la activación de los equipos de forma que se obtenga el valor de estado de vacío inicial, a partir del cual se puedan cuantificar posteriormente las modificaciones del campo magnético producidas por un elemento, y la consiguiente detección del mismo.

- Sensores infrarrojos (*Infrared, IR*): El principio de funcionamiento de estos dispositivos se basa en la capacidad de detectar y cuantificar la radiación electromagnética infrarroja reflejada por un cuerpo, el vehículo en este caso. En este sentido, el dispositivo incorpora un fototransistor capaz de medir incrementos de energía asociada a la radiación infrarroja, transformándola en una señal eléctrica que determina la presencia de un vehículo. Se pueden diferenciar dos tipos de sensores IR de interés: los pasivos y los activos.

  El sensor IR pasivo únicamente incorpora el fototransistor, de forma que solamente tiene la capacidad de recibir la radiación emitida por los objetos, mientras que el sensor IR activo incorpora, adicionalmente al fototransistor, un diodo LED infrarrojo que actúa como emisor. Al disponer de la capacidad de emitir una señal conocida, la respuesta es más fácilmente cuantificable, obteniendo mejores respuestas, especialmente en el empleo para detección de presencia de vehículos, dado que es posible tener una cierta independencia del nivel de luminosidad ambiental, que condiciona fuertemente los resultados en el caso del sensor pasivo (una gran incidencia de la luz del sol puede saturar el sensor, desvirtuando la medida ofrecida). La implantación de un sistema de este tipo implica dos pasos fundamentales:

  - Instalación de sensores: Para la instalación de estos sensores, se utilizan los dos métodos descritos para los sensores magnéticos, pero considerando que estos sensores deben encontrarse expuestos a la luz y, por lo tanto, no puede ser cubierta la parte superior de los mismos. Además, es aconsejable que la zona circundante no esté hundida con el objetivo de minimizar la posibilidad de acumulación de agua, suciedad o elementos que oculten la visión del sensor.
  - Calibración inicial. Tanto los sensores activos como los pasivos deben ajustarse a las condiciones iniciales con la plaza de estacionamiento libre. En este caso, para sensores pasivos los cambios en la luz ambiental (respecto a la de calibración) pueden conllevar falsas detecciones.

EE.UU. y Europa se encuentran entre los pioneros en la implementación de soluciones de aparcamiento inteligente. En San Francisco, el proyecto SFpark [SFpark], que comenzó en 2008, buscaba gestionar la ocupación de las plazas de aparcamiento en exteriores, así como generar un sistema de gestión integral para proveer a los ciudadanos con la disponibilidad de plazas de aparcamiento en tiempo real. Aunque en un principio este proyecto confió en la tecnología ofrecida por el proveedor Streetline [Streetline], finalmente acabó desistiendo de sus servicios e instalando los dispositivos provistos por la empresa Fybr [Fybr]. A nivel europeo, en 2012 Moscú adjudicó a la empresa WorldSensing [WorldSensing], la implantación de un sistema para la gestión de las plazas de aparcamiento en exteriores, cuyo despliegue se dividirá en tres fases, finalizando en 2014.

En la tabla siguiente se incluyen, algunas de las soluciones comerciales existentes, así como la tecnología utilizada por las mismas.

Tabla 6.1. Soluciones para detección de vehículos en exteriores

Fabricante	Tecnología	Instalación	Frecuencia de banda ISM	Batería
Nedap AVI [NEDAP]	Magnética e IR combinada o independiente según modelo seleccionado	Sensor insertado en pavimento o adherido a la superficie	868 MHz	5 años
Urbiotica U-SPOT [Urbiotica]	Magnética e IR	Sensor insertado en el pavimento	868 MHz	8 años
Tinynode [Tinynode]	Magnética	Sensor adherido a la superficie	868 MHz	10 años
Smartgrains ParkSense [Smartgrains]	Magnética	Sensor adherido a la superficie	868 MHz y 2,4 GHz	5 años (posibilidad de remplazarse)
Libelium [Libelium]	Magnética	Sensor insertado en asfalto	2,4 GHz	3 años

Dentro del marco del proyecto SmartSantander uno de los casos de uso se dirigió al despliegue de un sistema para gestión de plazas de aparcamiento en exteriores. Este sistema (compuesto por 400 dispositivos), como se explicará detalladamente a lo largo de este capítulo, utiliza la tecnología provista por Libelium para la detección de plazas libres. Adicionalmente a este sistema, y como parte de la fase final del proyecto se procederá a la instalación (finalizará en 2014) de la solución ofrecida por Nedap-AVI (alrededor de 300 dispositivos en la misma zona que la solución anterior), para la comparación tecnológica de ambas soluciones.

Finalmente, es interesante destacar que, mientras que las soluciones surgidas inicialmente tenían un enfoque principalmente orientado a la detección y monitorización, las soluciones actuales están ampliando su espectro de servicios incluyendo soluciones de identificación de vehículo, políticas de reserva de plaza o soluciones de pago adaptado en función de la ocupación de la plaza.

6.2 MODELO INICIAL

La arquitectura y topologías asociadas al despliegue del caso de uso de monitorización de las plazas de aparcamiento en exteriores [Galache11], se muestran a continuación en la figura 6.1. Como se puede derivar de la figura, los nodos de aparcamiento envían la información de estado (libre u ocupado) de la plaza, a través de los repetidores instalados en las farolas/fachadas, los cuales actúan como nodos retransmisores de la información enviada por los sensores de aparcamiento dentro de su área de cobertura, además de realizar y reenviar las medidas asociadas a la monitorización medioambiental fija. De esta forma, a través de la red de repetidores asociada a la monitorización ambiental estática, la información de plazas de aparcamiento es recibida en la pasarela correspondiente.

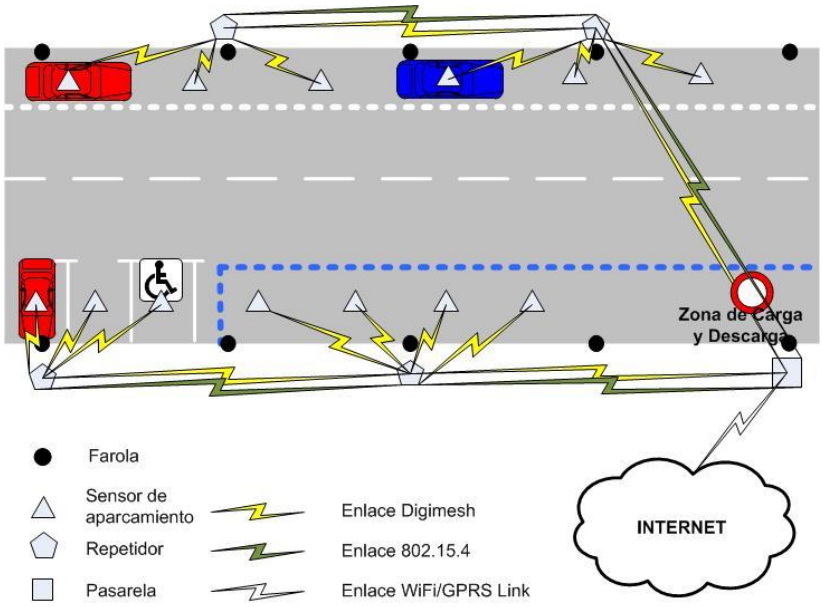

Figura 6.1. Arquitectura del servicio de gestión de aparcamiento en exteriores

Es importante indicar que a diferencia de los nodos repetidores, instalados en farolas/fachadas que se alimentan con baterías que se recargan durante la noche (aprovechando la corriente que alimenta la farola), funcionando de manera autónoma durante el día; o como ocurre en el caso de las pasarelas que se encuentran alimentadas de manera continua a través de un cable de red (mediante *PoE*), los nodos de aparcamiento se alimentan con baterías no recargables. Este hecho implica que el modo de operación de los nodos de aparcamiento ha de ser lo más eficiente posible en términos de consumo de batería. En este sentido, al contrario de repetidores/pasarelas que siempre se encuentran operativos, los nodos de aparcamiento deben manejar los correspondientes mecanismos de ahorro de consumo (estados de hibernación/sueño, transmisiones acotadas, etc.). Esto implica que los nodos de aparcamiento no operan como nodos retransmisores de la información, sino que se configuran como elementos finales que sólo transmiten/reciben información dirigida a ellos, sin reenviar información enviada/procedente de otros nodos. Además, debido a las restricciones en términos de batería, la experimentación a nivel de nodo no se habilita con lo que, como se indicó con anterioridad, solamente están provistos de una única interfaz radio implementando el protocolo *Digimesh* que permite establecer las rutas correspondientes para el envío de la información asociada al servicio de gestión de las plazas de aparcamiento. En función de los repetidores de los que dependan, los nodos de aparcamiento se asociarán con uno u otro *cluster*, de forma que se encuentran configurados para enviar toda la información a la correspondiente pasarela. Respecto a la gestión de la red, aunque estén provistos de la interfaz *Digimesh*, los nodos no podrán reprogramarse de manera remota, ni tendrán la capacidad de envío/recepción de comandos, puesto que sólo se encontrarán operativos cuando envíen una trama de información sobre el estado de ocupación de una plaza, permaneciendo en estado de hibernación el resto del tiempo. Este modo de funcionamiento se traduce en una reducción del consumo de la batería y, por tanto, un aumento de la vida útil de la misma, manteniendo operativo el servicio durante más tiempo y pudiendo alargar los ciclos de sustitución de baterías.

En la siguiente figura se muestran la principal zona de la ciudad donde se han instalado la gran mayoría de los sensores desplegados (así como donde se instalará la nueva solución provista por NEDAP):

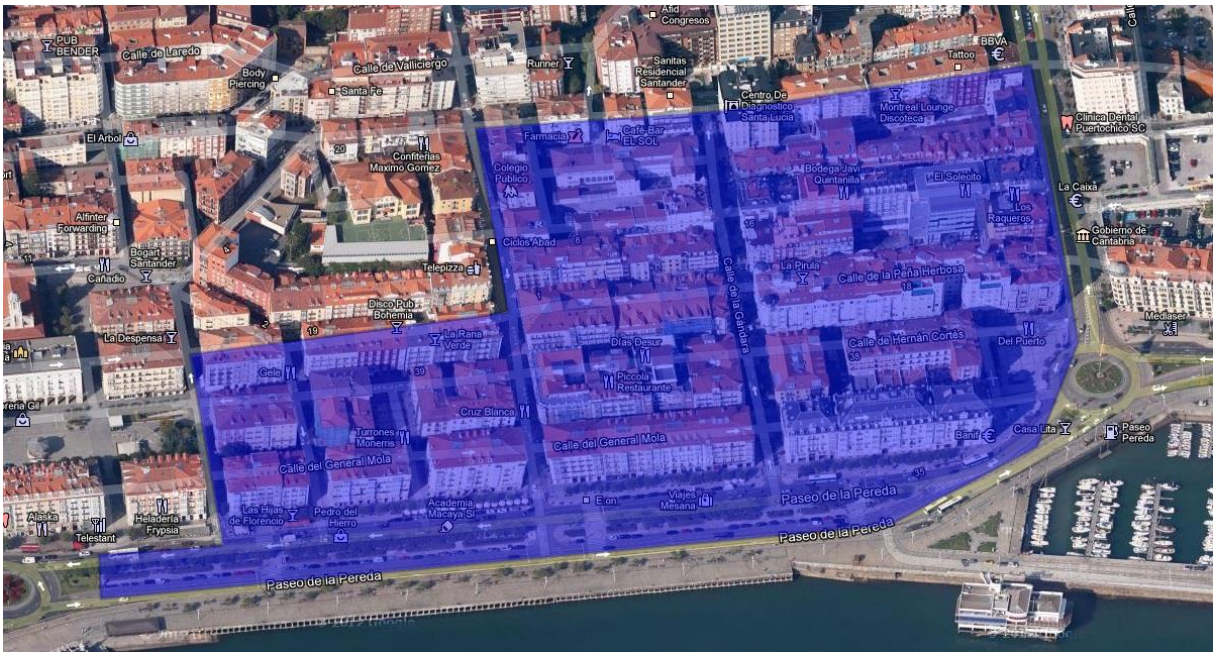

Figura 6.2. Zona de instalación de los sensores de aparcamiento

Las calles que se indican en la figura 6.2, se encuentran en el centro de la ciudad de Santander, perteneciendo todas las plazas donde se realiza la instalación de los sensores a la zona de OLA (Ordenanza Limitadora de Aparcamiento). Esta zona se caracteriza por la limitación de estacionamiento de los vehículos en horario de mañana (de 10 a 14 horas) y de tarde (de 16 a 20 horas) de los días laborables, así como los sábados en horario de mañana, durante un máximo de dos horas. Una vez superado el tiempo de estacionamiento máximo de dos horas, el vehículo debe ser estacionado en otra zona de OLA (cada zona implica un determinado conjunto de plazas de aparcamiento).

En la siguiente figura se muestra un detalle de los sensores de aparcamiento instalados:

Figura 6.3. Detalle de instalación de los sensores de aparcamiento

Como se observa, los nodos de aparcamiento se encuentran enterrados en el pavimento, para lo que se realiza un agujero (10 cm de profundidad y 12 cm de diámetro) en el asfalto, donde se introduce una cápsula dentro de la que se encuentra el nodo correspondiente junto con la batería (parte izquierda de la figura). Dentro de la cápsula se introducen unas bolsas de sílice, cuyo cometido es absorber la humedad que se pueda producir dentro de la cápsula (y evitar así la condensación). El nodo se coloca de forma que la antena se encuentre en el lugar más cercano a la tapa y a la

superficie del pavimento, con la finalidad de intentar mejorar las condiciones de propagación de la señal. La cápsula se sella con una resina hidroexpansiva que permite impermeabilizar el interior de la misma, extendiéndose una capa de hormigón sobre la tapa que permite absorber los esfuerzos realizados por el paso de los neumáticos de los vehículos, y además como se puede observar en la figura (centro), al cabo de unos días tiende a mimetizarse con el aspecto del asfalto, minimizando el impacto visual así como las posibles acciones asociadas con el vandalismo.

Una vez definida la forma en la que se lleva a cabo la comunicación, se detalla a continuación el funcionamiento del protocolo de detección (figura 6.4):

- Como ya se indicó en la descripción del estado del arte, el funcionamiento de los sensores de detección de aparcamiento instalados se basa en un sensor ferromagnético. el cuál mide las variaciones del campo magnético terrestre en los tres ejes cartesianos x, y, z. De esta forma, la presencia de los elementos metálicos que componen la estructura de un vehículo provoca la variación del campo magnético terrestre, que es detectada por el sensor. En aras de evitar falsas detecciones debidas a coches aparcados en las plazas adyacentes, se define un umbral que permite discernir la presencia de vehículo en la plaza, desechando falsas detecciones asociadas a vehículos en plazas contiguas. Lógicamente, y principalmente en aquellos zonas en las que las plazas de aparcamiento no se encuentran delimitadas (por ejemplo, aparcamientos en hilera sin delimitación de plazas), pueden producirse lecturas erróneas en situaciones en las que alguno de los vehículos no estacione correctamente sobre uno de los sensores.

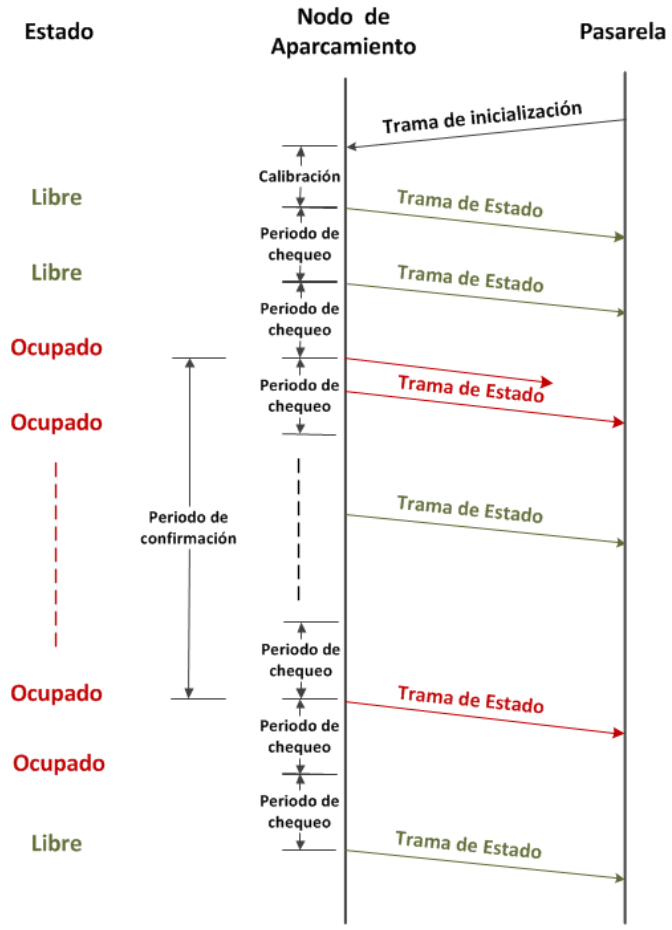

Figura 6.4. Funcionamiento del protocolo de detección de vehículos

- En el momento de la instalación, los nodos se calibran con la plaza vacía de forma que tomen un valor de referencia que se corresponde con el estado libre de la plaza. Para ello, una vez instalado y cubierto el sensor, se le envía una trama para reiniciarlo y que comience el proceso de calibración. La detección de vehículo se realiza restando el valor medido (considerando los tres ejes) con el valor de calibración, comparando el valor obtenido con el umbral de detección (si se supera se asigna el estado de ocupado a la plaza).
- Una vez calibrado, el nodo se duerme durante un período fijo establecido (5 minutos en este caso), deshabilitando tanto el módulo radio como la placa de sensores (estado *deepsleep*), para evitar un consumo excesivo de batería. Pasado el período de hibernación, el nodo se despierta, enciende la placa de sensores y comprueba el estado de la plaza. Si el estado de la plaza supone un cambio, ya sea de libre a ocupado o viceversa, se enciende el módulo radio y se envía a la pasarela correspondiente, la trama de estado de la plaza, así como los valores de variación del campo magnético recogidos. En caso de que la plaza se mantenga en el mismo estado, el comportamiento es el siguiente:
  - Si el estado se mantiene ocupado, entonces no se envía el paquete de estado correspondiente, asociando en la pasarela la ausencia de recepción como estado ocupado. Este comportamiento responde a las posibles pérdidas de comunicación asociadas con la ubicación de los vehículos en ciertas posiciones sobre el nodo, que pueden producir la pérdida de uno o varios paquetes enviados desde el mismo. En aras de discernir entre si la plaza se encuentra ocupada o si el nodo se ha quedado inactivo, se fija un período de confirmación (actualmente 60 minutos) enviándose el estado ocupado del nodo (aunque éste no haya cambiado).
  - Si el estado se mantiene libre, se envía siempre el paquete con la información de estado puesto que, en principio, la probabilidad de pérdida de paquetes se reduce respecto al estado ocupado. De esta forma, cuando el servidor recibe periódicamente (cada 5 minutos) paquete de estado de un determinado nodo, mantendrá el estado de la plaza correspondiente como libre.

  Este problema de comunicación previamente citado, tiene mayor incidencia a la frecuencia de 2.4 GHZ (frecuencia a la que trabajan la solución de Libelium), que a 868 MHZ (frecuencia a la que trabajan otras soluciones como NEDAP-AVI). Sin embargo, en el momento de la instalación de la solución de Libelium, el protocolo *Digimesh* no era soportado a 868MHz, lo que suponía tener que realizar un protocolo de enrutamiento sobre los módulos radio para poder asegurar la provisión de servicio y la gestión de la red. Adicionalmente, la banda de 868 MHz sólo permite 1 canal, en lugar de los 12 asociados a la banda de 2.4 GHz, con lo que los *clusters* adyacentes no se podrían configurar en diferentes frecuencias, pudiendo aumentar las interferencias en la red.

- En cualquiera de las situaciones (descritas anteriormente) en las que el nodo tiene que enviar un paquete de estado, se realizan tres intentos (cada uno incluyendo los 3 intentos a nivel MAC descritos en el Capítulo 4) de envío (separados 1 segundo), para tratar de asegurar que el paquete es recibido correctamente en la pasarela. Si el paquete no ha sido enviado correctamente (no se recibe reconocimiento a nivel MAC), se repite el proceso (hasta 3 veces) espaciado cada minuto.

Teniendo en cuenta el protocolo descrito con anterioridad, el tiempo de duración de las baterías oscilará entre 2 y 3 años, dependiendo de la cobertura de los nodos (implica mayor o menor número

de retransmisiones), con respecto a su repetidor correspondiente. Dada la ubicación de las baterías dentro de las cápsulas, éstas no podrán ser sustituidas de manera sencilla, de forma que se reemplazarán los nodos de manera completa (intentando coincidir con los ciclos de renovación del pavimento). Salvo la batería que debe ser sustituida, tanto la placa *waspmote*, como el módulo *Digimesh* y la placa de medida, pueden reutilizarse una vez cambiada la batería.

6.3 PROTOCOLO ADAPTATIVO

El protocolo descrito en el apartado anterior presenta una serie de limitaciones, tanto a nivel de provisión de servicio, como de gestión del dispositivo. Respecto a la provisión de servicio, el funcionamiento definido se encuentra encorsetado, presentando unos parámetros de configuración (umbral de decisión, período de muestreo, número de retransmisiones) fijados al comienzo del protocolo, y que no pueden ser modificados dinámicamente una vez que los nodos se encuentran instalados. En lo que respecta a la gestión de los dispositivos, según se comentó con anterioridad, éstos no pueden ser accesibles para la reprogramación o el envío/recepción de comandos desde la pasarela porque, salvo en el momento de envío de datos, el tiempo restante se encuentran en estado de hibernación. Para tratar de abordar y solucionar ambas problemáticas se ha modificado el protocolo anterior, buscando una funcionalidad más flexible y adaptativa, tanto para la provisión de servicio como para la gestión de los dispositivos.

- Ventana de recepción de comandos: Como se ha señalado, por razones de consumo de batería los nodos no pueden estar constantemente despiertos para recibir los comandos enviados por la pasarela (reprogramación, configuración remota). Para solventar este problema, el nodo implementa la capacidad de habilitar de manera periódica una ventana temporal durante la cual se encuentra activo y, por lo tanto puede recibir y procesar cualquier tipo de comando enviado por la pasarela.

  Como se puede observar en la figura 6.6, antes de abrir la ventana de recepción de comandos, el nodo envía un mensaje a la pasarela indicando que la ventana está abierta y que durante la duración de la misma el nodo es capaz de recibir un comando desde la pasarela. Si la pasarela tiene algún comando para enviar, ya sea referente a la provisión de servicio o a la gestión del dispositivo (principalmente reprogramación) lo enviará, cerrándose la ventana cuando finalice el comando correspondiente. En caso contrario, una vez expirado el temporizador de la ventana, ésta se cerrará y el nodo volverá al funcionamiento correspondiente al protocolo de gestión de aparcamiento. Tanto el valor de tiempo en el que la ventana se encuentra abierta (tiempo de ventana), como el período entre dos aperturas de ventana (período de ventana), son parámetros configurables y modificables en tiempo de ejecución, como se detallará más adelante.

  Los paquetes de apertura de ventana y reconocimiento de comando, se muestran a continuación:

Figura 6.5. Mensaje de apertura de ventana (izquierda) y reconocimiento de comando (derecha)

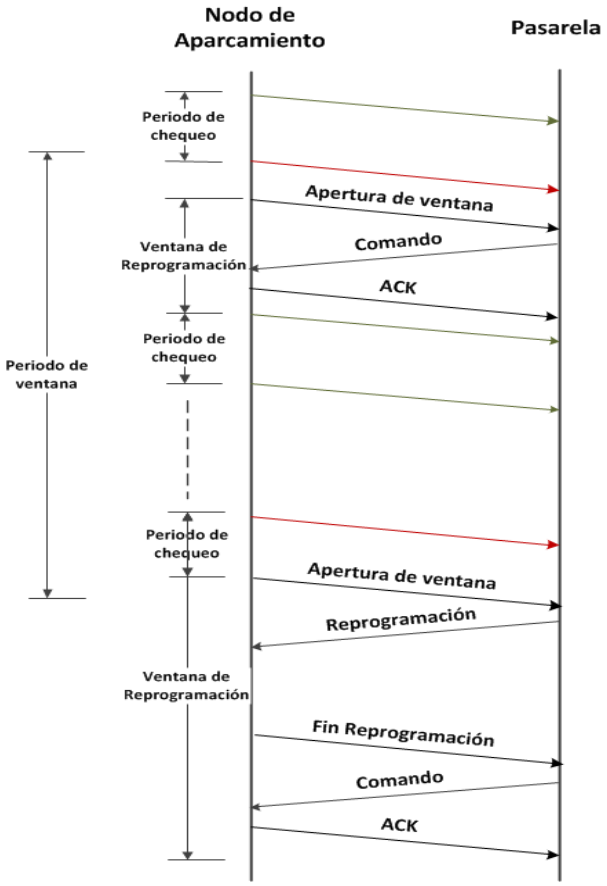

Figura 6.6. Detalle de apertura de ventana de acceso al nodo

Como se observa en la figura 6.5, tanto el paquete de apertura de ventana, como el reconocimiento de un comando, contienen el *identificador de la Carga* para que sean recibidos y procesados correctamente por la pasarela, así como sus correspondientes identificador de ventana para indicar a la pasarela que el nodo acaba de abrir su ventana de recepción, y el identificador de comando del que se está reconociendo su correcta recepción en el nodo, respectivamente.

- Provisión de servicio adaptativa: En aras de adaptar la funcionalidad de detección a una plaza de aparcamiento, ciertos parámetros relacionados con el protocolo pueden ser variadas durante el tiempo de ejecución del protocolo. A continuación se definen los parámetros configurables remotamente:
  - Vectores de detección de libre y ocupado: En lugar de chequear el estado de la plaza cada un cierto intervalo fijo (5 minutos en el protocolo original), se definen los diez primeros intervalos que se utilizarán para comprobar el estado de la plaza, diferenciando si ésta se encuentra libre u ocupada. Esta diferenciación se debe a que, teniendo en cuenta la alta ocupación de las plazas que se monitorizan, el comportamiento lógico se correspondería con la ocupación de una plaza en un corto espacio de tiempo tras quedarse vacía y, una vez ocupada, ésta permaneciese en este estado durante un espacio de tiempo más amplio. En este sentido, se define un vector de intervalos decreciente en el caso ocupado, de forma que una vez ocupada la plaza el siguiente chequeo de su estado sea un tiempo elevado (por ejemplo 10 o 15 minutos), disminuyendo el intervalo de chequeo para las siguientes comprobaciones (más probabilidades de que la plaza quede libre). Por el contrario, si la plaza

se encuentra libre, las posibilidades de que se ocupe en un corto espacio de tiempo son bastante altas y, por lo tanto, el vector de intervalos en estado libre presentará un comportamiento creciente, por ejemplo los primeros intervalos a 1 minuto, aumentando después, puesto que si la plaza no se ocupa inmediatamente quizás estamos en un período de baja ocupación (por ejemplo de noche o fuera del horario laboral).

- Intervalo continúo de libre y ocupado: Una vez que el estado (libre u ocupado) se mantiene en los diez primeros intervalos de chequeo (intervalos más críticos para que se produzca el cambio de estado), se establece un intervalo fijo para el resto de las comprobaciones de estado. En principio, y siguiendo con la tendencia previamente indicada, el tiempo de estado libre sería superior que el de estado ocupado, pero en este estado estacional ambos tenderán a un valor similar.
- Modo continúo: Este modo establece como intervalo fijo de detección un valor de 1 minuto y, además, fija un funcionamiento para el estado ocupado igual que para el libre, de forma que cada minuto siempre se envía la trama de estado (aunque el estado sea ocupado y no haya habido cambio). Este modo de funcionamiento permite realizar un seguimiento de la plaza muy preciso, de forma que se pueden detectar todos los cambios de estado de la misma. Con el intervalo de 5 minutos si, por ejemplo, una plaza quedase libre y se ocupase en menos de 5 minutos, no se detectaría el cambio de estado a libre, sino que se consideraría que el estado se ha mantenido a ocupado. Sin embargo, este funcionamiento en modo continuo, no se puede mantener de manera constante porque supone un elevado consumo de batería, con la consiguiente disminución en el tiempo de vida de los sensores. La finalidad de este modo es, por lo tanto, estimar el funcionamiento de cada una de las plazas (en este caso será similar porque están en la misma zona), adaptando tanto los vectores de intervalos, como los intervalos continuos para los estados de libre y ocupado, asegurando una detección más eficiente y adaptada a cada una de las plazas/zonas.
- Umbral de detección: En principio este valor no debería modificarse, pero en función de los resultados de detección (calibración del sensor, distintas condiciones del entorno), también se podría modificar para obtener una detección más precisa.
- Número máximo de retransmisiones: En principio, como se comentó anteriormente, este parámetro está fijado a un valor de 3, pero se puede variar para adaptarlo en función de las condiciones del entorno radio, asegurando así la correcta recepción de los mensajes de estado enviados por los nodos.
- Tiempo periódico de envío de estado: Como se indicó en la descripción del protocolo, cuando el nodo se encuentra en estado ocupado no envía cada vez que se comprueba el estado (si este se mantiene en ocupado), sino que envía cada cierto intervalo periódico para indicar que el nodo sigue operativo.

Todos estos parámetros, así como el tiempo y el período de ventana son configurables mediante el siguiente paquete:

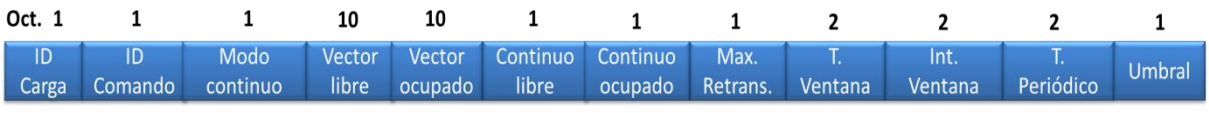

Figura 6.7. Detalle del paquete de configuración

El paquete mostrado en la figura 6.7, es enviado por la pasarela cuando recibe el mensaje de la apertura de la ventana de reprogramación por parte del nodo. Como se comprueba fácilmente contiene todos los parámetros configurables previamente descritos, así como el tiempo e intervalo de ventana. Una vez que el nodo recibe el paquete de configuración, envía un reconocimiento (figura 6.5) a la pasarela de que lo ha recibido correctamente, asignando los nuevos valores recibidos a los parámetros correspondientes.

- Capacidad de reprogramación: La apertura de una ventana de reprogramación no sólo permite la recepción de comandos específicos (como el de configuración), sino que ofrece la posibilidad de reprogramar el nodo remotamente. Es este sentido, aunque el tamaño de la ventana es inferior al tiempo de reprogramación, si el nodo entra en estado de reprogramación la ventana permanece abierta (interfaz *Digimesh* activa), hasta que el período de reprogramación termine (figura 6.6). De hecho, como con alta frecuencia, un proceso de reprogramación suele acompañarse de la consiguiente ejecución del programa cargado, una vez terminada la reprogramación del nodo el temporizador de ventana vuelve a empezar de cero para permitir la recepción de otra trama adicional, ya sea asociada a la reprogramación (descritas en el Capítulo 4), o a la provisión de servicio (mensaje de configuración). Este modo de operación permite poder enviar tantos comandos como sea necesario en una misma ventana, sin necesidad de esperar a la apertura de la siguiente. De esta forma, se puede asignar un intervalo de tiempo bajo a la ventana (suficiente para enviar el comando de apertura y recibir el correspondiente comando/s de la pasarela), de forma que se reduzca el consumo de batería al máximo cuando no haya ningún comando para enviar por parte de la pasarela.

El protocolo adaptativo descrito en esta sección se ha probado exhaustivamente en el *cluster* de la universidad donde se encuentran instalados 10 sensores como despliegue piloto y, posteriormente, se ha instalado en dos nodos presentes en el centro de la ciudad para observar el comportamiento en esta zona. Actualmente, los nodos se encuentran funcionando en modo continuo recogiendo valores del estado de ocupación de las plazas de la manera más precisa posible para, una vez analizados estos valores, poder inferir patrones de ocupación de las plazas y configurar los parámetros descritos correspondientemente, así como derivar datos de consumo y de comunicación (pérdida de paquetes, número de retransmisiones).

6.4 CONCLUSIONES

En este capítulo se ha trabajado sobre la mejora de uno de los casos de uso, la monitorización de las plazas de aparcamiento en exteriores desplegado en la ciudad de Santander. En este sentido, se ha descrito el proceso de instalación, la zona donde se lleva a cabo el despliegue, así como el funcionamiento del protocolo utilizado para la detección del estado de ocupación de una plaza de aparcamiento.

Con la finalidad de mejorar el servicio ofrecido, tanto a nivel de servicio como de gestión de los dispositivos, se ha definido e implementado un protocolo adaptativo que permite configurar el nodo de una manera dinámica en función de las particularidades del entorno donde se encuentre instalado. Este nuevo protocolo está instalado actualmente en algunos dispositivos, con la finalidad de comprobar su rendimiento en relación con el protocolo inicial.

7 CONCLUSIONES Y LÍNEAS FUTURAS

Llegados aquí, es el momento de exponer las conclusiones derivadas de los distintos bloques temáticos que se han analizado, y que cubren desde la gestión uniforme de las redes de sensores hasta la concepción de una arquitectura que da soporte a los planos de experimentación, servicio y gestión de la red.

Finalmente, se concluye presentando las líneas futuras entre las que destacan, la federación de infraestructuras de experimentación, la utilización de métodos de computación basados en la nube y la simbiosis entre el paradigma IoT y las redes sociales.

7.1 CONCLUSIONES

Las ciudades inteligentes [ESMARTCITY] se sitúan como el punto de encuentro entre el soporte, la participación y el compromiso del usuario con iniciativas innovadoras basadas en las TIC y fomentados por los denominados laboratorios vivos; junto con la creación y provisión de redes experimentales a gran escala (algunas de ellas promovidas dentro de la iniciativa FIRE), consideradas como excelentes campos de prueba para la experimentación y la investigación dentro de la FI. En este sentido, tanto para recoger una gran parte de la información asociada a estos entornos de ciudades inteligentes, como para conformar la red masiva sobre la que realizar la experimentación FIRE asociada a estos entornos de ciudades inteligentes, las WSNs se erigen como un elemento clave.

El trabajo desarrollado dentro de esta Tesis Doctoral ha cubierto la evolución desde las WSNs, como fuentes de información de medida básica, a priori difíciles de gestionar y con capacidades de experimentación limitadas, a las ciudades inteligentes en las que se aplica el concepto de la medida de diferentes parámetros a la mejora de los servicios urbanos, y a la creación de aplicaciones dirigidas a los ciudadanos, ofreciendo a la vez bancos de prueba a la comunidad científica. En definitiva, bajo el paradigma de las ciudades inteligentes se acomoda una amplia pléyade de actores, abarcando desde los experimentadores, pertenecientes a la comunidad científica, a los proveedores de servicio y desarrolladores de aplicaciones, pasando por las Administraciones Públicas y los ciudadanos, todos ellos bajo la supervisión de los gestores de la red.

En lo referente a la gestión y manejo de las redes de sensores inalámbricas, dentro de este trabajo se han presentado diferentes implementaciones de capas de adaptación y abstracción de interfaces subyacentes heterogéneas, conocidas como *middleware*, y que permiten la gestión de una manera homogénea y uniforme de un despliegue heterogéneo. Entre ellas, como resultado de varios proyectos de índole nacional e internacional se destacan la ULLA y la GLL/PLAMIN.

La aportación clave de este trabajo se encuadra dentro del proyecto SmartSantanter, el cuál persigue la implantación de un despliegue masivo de dispositivos IoT en el entorno de la ciudad de Santander, en la que se han desplegado más de 12.000 sensores. Este despliegue, caracterizado por la convivencia de diferentes tecnologías, se compone de alrededor de 3.000 dispositivos IEEE802.15.4, 200 dispositivos provistos de módulo GPRS/3G, 4.000 usuarios móviles (sensores asociados a un teléfono móvil) y 2.600 etiquetas duales dotadas de interface NFC y código QR, desplegadas tanto en ubicaciones estáticas (farolas, fachadas, paradas de autobuses), como embarcadas en vehículos móviles (autobuses, taxis). Por otro lado, la plataforma desplegada ofrece una amplia capacidad de medición de parámetros medioambientales, aparcamiento, tráfico, riego, que sirven como base para el desarrollo de diferentes servicios/aplicaciones desarrollados dentro del marco del proyecto y que, a su vez, se ofrecen a terceras partes (proveedores de servicio, desarrolladores de aplicaciones) para la generación de nuevos servicios de valor añadido. Los servicios ya desplegados se asocian a los casos de uso abordados dentro del proyecto: monitorización medioambiental fija y móvil, gestión de aparcamiento en exteriores, riego inteligente y monitorización de tráfico, así como dos aplicaciones para entornos móviles: realidad aumentada y sensado participativo.

En este contexto, se ha definido la arquitectura que posibilita la gestión del despliegue realizado, tanto a alto como a bajo nivel, la cual se caracteriza por presentar un enfoque genérico para un despliegue IoT sintonizado a las necesidades de una ciudad inteligente, y concebido en base a una

jerarquía de tres niveles, a saber: nodo IoT, pasarela y servidor. De esta forma, los nodos IoT desplegados (la mayoría caracterizados por una capacidad de cómputo limitada), se agrupan formando los denominados *clusters* que se encuentran controlados por una pasarela. Esta pasarela se encarga de enviar los datos recogidos por los sensores hacia el servidor, así como se erige como punto intermedio para las comunicaciones (principalmente asociadas a la gestión de la red) desde el servidor hacia los nodos IoT. Por último, en el nodo servidor se ubican todas las entidades (bases de datos, aplicaciones) que se encargan de la provisión de los diferentes servicios y experimentos asociados al despliegue correspondiente, así como de la gestión del mismo de manera remota.

Es importante resaltar que la arquitectura utilizada como modelo de referencia para la infraestructura desplegada no confía únicamente en despliegues particulares asociados a un servicio específico, sino que también provee un entorno de experimentación urbano para proveedores de servicios y de tecnología. Aunque cabe reconocer que la importancia de las capacidades de experimentación en la arquitectura de una ciudad inteligente (madura en el futuro), podrían quedar en un segundo plano, las incipientes ciudades inteligentes actuales se beneficiarán de la capacidad para experimentar erigiéndose en plataforma de lanzamiento y evaluación para el desarrollo de futuros servicios. Adicionalmente, la arquitectura desarrollada presenta unas características fácilmente exportables, convirtiendo el enfoque desarrollado en una arquitectura que se puede replicar e implantar en otras ciudades, de manera sencilla.

A partir de los tres ejes principales del proyecto, a saber: provisión de servicio, capacidad de experimentación y gestión remota, en el marco de esta Tesis Doctoral, se ha profundizado en cada uno de ellos, en aras de validar la arquitectura desarrollada, en términos de manejabilidad para gestionar de manera eficiente los recursos desplegados, de operatividad para realizar experimentos sobre ellos y de flexibilidad para añadir servicios nuevos o mejorados, respecto a los ya desplegados dentro de los casos de uso desarrollados dentro del proyecto.

Desde el punto de vista de la gestión del servicio, se ha implementado un protocolo de envío de comandos y reprogramación de los nodos de manera remota, adaptado a las peculiaridades de un despliegue masivo dentro de un escenario urbano. Para ello, considerando las limitaciones de las implementaciones de reprogramación remota actuales, acotadas en su mayoría a entornos poco hostiles y despliegues de bajo número de dispositivos (principalmente en interiores), se ha realizado una implementación más robusta que, principalmente, presenta en el receptor (nodo IoT), una gestión más eficaz de los paquetes perdidos y recibidos fuera de secuencia, mientras que en el servidor (pasarela) se desarrolla un mecanismo de retransmisión más eficiente. Unido a estas características, y teniendo en cuenta la particularidad del despliegue realizado, en el que provisión de servicio y capacidad de experimentación coexisten de manera simultánea con la gestión de la red, tanto el servidor a través de un multiplexor de puertos, como el nodo mediante el uso del *watchdog*, habilitan este comportamiento concurrente.

En lo referente a la capacidad de experimentación (a nivel de nodo), se ha diseñado, implementado y validado un protocolo para el descubrimiento de vecinos, utilizando la interfaz de experimentación (802.15.4 nativa) provista por los nodos. Desde el punto de vista de la ejecución del experimento, se accede a la plataforma (autenticación y autorización), se reservan los nodos correspondientes (asociados a varias pasarelas) y durante el tiempo necesario para realizar el experimento, se reprograman los nodos reservados mediante el protocolo de reprogramación remota implementado

y, finalmente, se envían los correspondientes resultados del experimento a cada uno de los experimentadores. Este ejemplo de experimentación sobre la plataforma desplegada, muestra la enorme potencialidad de la misma, permitiendo extender desarrollos específicos de laboratorio a implementaciones sobre un despliegue masivo en un entorno real.

Respecto a la provisión de servicio, se ha trabajado en la mejora del caso de uso de la gestión de aparcamientos en exteriores, presentando un servicio de gestión adaptativa para la detección de la ocupación de plazas de aparcamiento. Por un lado, se ha diseñado un protocolo con intervalos de tiempo variables para adaptarse de una manera más eficiente a los cambios de estado libre-ocupado y ocupado-libre; mientras que por el otro lado, se han habilitado ventanas para el envío/recepción de comandos y la reprogramación remota de los nodos enterrados, permitiendo una mayor flexibilidad del comportamiento de los mismos, así como, aunque limitada, una gestión remota de estos nodos. Finalmente, teniendo en consideración la capacidad, por parte de la plataforma desarrollada, para la inclusión de nuevos servicios de manera sencilla, la gestión de aparcamiento adaptativa opera de manera paralela y concurrente al resto de servicios desplegados.

7.2 LÍNEAS FUTURAS

La infraestructura desplegada en SmartSantander se ha convertido en un instrumento esencial hacia el liderazgo europeo en tecnologías habilitadoras para IoT, y para la provisión de una plataforma única de estas características a la comunidad científica, adecuada para la experimentación a gran escala y la evaluación de conceptos IoT bajo condiciones reales. Para ofrecer la correspondiente capacidad de experimentación y provisión de servicio, tanto la plataforma desplegada como la arquitectura definida constituyen una propuesta innovadora armoniosamente integrada en el entorno urbano. En esta línea, y debido a la constante transformación y necesidad de modernización de los entornos urbanos, continuamente se presentan líneas de mejora para adaptarse a los nuevos requerimientos y necesidades que demanden, principalmente, los ciudadanos y las Administraciones Públicas. Muchas de estas líneas, se están abordando dentro del Grupo de Ingeniería Telemática, como continuación de SmartSantander y dentro del marco de proyectos de índole nacional y europea. A continuación, se detallan las más relevantes:

- Federación de plataformas: Actualmente, bajo el paraguas de diferentes iniciativas y proyectos a nivel mundial, se están llevando a cabo numerosos despliegues masivos de dispositivos IoT. Por este motivo, adquiere gran importancia la federación de todos estos despliegues, permitiendo a los usuarios finales (proveedores de servicio, desarrolladores de aplicaciones, experimentadores), un acceso unificado a todos los recursos, independientemente del despliegue (localización física) al que pertenezcan. En este sentido, el proyecto FP7 FED4FIRE [FED4FIRE] se concibe para ofrecer, de manera sencilla y abierta, las facilidades desarrolladas bajo el paraguas FIRE, a todos los experimentadores; ofreciéndoles herramientas comunes para ejecutar experimentos innovadores independientemente del despliegue en el que se encuentre ubicado el nodo.
- Combinación de los datos IoT con la información de las redes sociales: Teniendo en consideración el imparable crecimiento de los despliegues de redes de sensores y de la capacidad de medida de un mayor número de parámetros por parte de los mismos, así como el aumento en el uso de las redes sociales, se ha de trabajar en la interacción de la información proveniente de ambas

fuentes. Dentro del proyecto ICT-PSP RADICAL [RADICAL], se abordará esta interacción, habilitando el desarrollo y despliegue de innovadores servicios interoperables de multi-medida y perspectiva social.

- Capacidad de replicar la plataforma: De manera progresiva, cada día aumenta el número de ciudades que buscan realizar despliegues y desarrollar plataformas que permiten una gestión eficiente de los recursos urbanos, así como una mejora de la calidad de vida de los ciudadanos. En este sentido, cobra especial importancia habilitar los mecanismos y las metodologías necesarias para replicar, de manera sencilla, el modelo desarrollado en una ciudad en otras ciudades. La plataforma desarrollada dentro del proyecto RADICAL se centra, precisamente, en facilitar un gobierno inteligente así como una capacidad de replicación sencilla de los servicios asociados a diferentes ciudades y regiones, mediante mecanismos que evalúen las peculiaridades de cada ciudad en términos de sus infraestructuras técnicas, las características socio-económicas y los condicionantes legales.
- Mejora de la plataforma SmartSantander: El despliegue realizado incluye dispositivos de medida de parámetros diversos, para cada uno de los cuales existe una gran gama de dispositivos de diferentes características y precisión, con lo que se pueden añadir diferentes capacidades adicionales a la plataforma desplegada, tanto a nivel *hardware* (añadiendo nuevos dispositivos de medida), como a nivel *software* (implementando nuevas técnicas de procesado de datos). En este sentido, el proyecto europeo FP7 EAR-IT [Hollosi13], tiene como uno de sus objetivos la realización, sobre la plataforma ofrecida por SmartSantander, de un experimento relacionado con el uso de acústica inteligente destinada a soportar aplicaciones de alto valor social, fomentando la innovación y la sostenibilidad. En particular, el despliegue de sensores de ruido realizado dentro del proyecto SmartSantander será complementado con la adición de nuevo *hardware* más específico en la detección de ruido, así como de la realización de desarrollos a nivel *software* que permitirán un tratamiento más específico de la información generada; consiguiendo de esta forma, mejorar el rendimiento completo de la red en lo referente a la detección de ruido.
- Inclusión de técnicas de computación en la nube: La continua proliferación de despliegues de dispositivos IoT, así como la capacidad de los mismos para medir un gran número de parámetros con bajos períodos de muestreo, se traduce en la generación de ingentes cantidades de información que deben ser almacenadas y gestionadas adecuadamente, no siendo las herramientas actuales (por ejemplo, las tradicionales soluciones para almacenamiento de datos basados en *SQL,* las más apropiadas y teniendo que recurrir a herramientas más sofisticadas (por ejemplo, servicios de almacenamiento basados en la nube). Con este propósito, el proyecto ClouT [ClouT], encuadrado dentro de la llamada conjunta Unión Europea-Japón, lanzada por la Comisión Europea dentro del FP7, desarrolla como concepto global la utilización de la computación en la Nube como habilitador para conectar la Internet de las cosas con la Internet de las Personas a través de la Internet de los servicios; así como para establecer una plataforma de colaboración y comunicación eficiente, explotando todas las posibles fuentes de información. ClouT proveerá infraestructuras, servicios, herramientas y aplicaciones que serán reutilizadas por diferentes actores de la ciudad tales como ayuntamientos, ciudadanos, desarrolladores de servicio e integradores de aplicaciones, para crear, desplegar y gestionar aplicaciones centradas en el usuario, y beneficiándose de los últimos avances en la Internet de las cosas y los dominios de la Nube.

- Medición e impacto de los campos de radiación electromagnética: Como ha sido indicado por la Organización Mundial de la Salud [OMS11], un amplio número de estudios se ha realizado en las últimas dos décadas para evaluar si los teléfonos móviles suponen un riesgo potencial para la salud. A fecha actual, ningún efecto adverso sobre la salud ha sido establecido como consecuencia del uso del teléfono móvil. No obstante, para proteger al público contra los efectos conocidos sobre la salud de los campos electromagnéticos, ciertos límites han sido establecidos por el Consejo Europeo. A pesar de estos límites de protección, las redes inalámbricas de área personal, local y extensa (redes de acceso celular) generan campos electromagnéticos (*ElectroMagnetic Fields, EMFs*), que inducen preguntas, controversias y preocupaciones crecientes en la población. En este sentido, el proyecto Lexnet (*Low EMF Exposure Networks*) [Lexnet] propone desarrollar mecanismos efectivos para reducir en un 50 % (al menos), el grado de exposición pública a la EMF, sin comprometer la calidad de servicio, desplegando nuevos dispositivos que permiten la medida del grado de exposición a la EMF en las principales bandas de frecuencia, y definiendo un índice global de exposición para todos en el espacio y en el tiempo.

Además de las líneas futuras en el ámbito de las ciudades inteligentes, basadas en el despliegue realizado en el proyecto SmartSantander y abordadas por estos nuevos proyectos europeos, dentro del proyecto SmartSantander, se está trabajando en líneas adicionales, principalmente encaminadas a mejorar la gestión de la red desplegada. Los principales puntos a destacar, se concentran en desarrollos asociados con la seguridad de la plataforma, la interacción entre nodos móviles y fijos y la mejora tecnológica de la plataforma, como se muestra a continuación:

- Seguridad de la plataforma: En la actualidad, las plataformas de sensores, como dispositivos embebidos de baja capacidad y asociados a entornos acotados y controlados, incorporan técnicas criptográficas a nivel físico y MAC, tales como AES128 [AES128]. En aras de mejorar la seguridad en la red, se está trabajando en la implantación de diferentes mecanismos de seguridad adicionales: técnicas de criptografía simétrica con cadenas de llaves basadas en resúmenes criptográficos, criptografía basada en identidad utilizando curvas elípticas y métodos de criptografía de clave pública asociada a certificados. De esta manera, se intenta asignar diferentes niveles de clave: entre nodo y servidor, entre nodos vecinos, a nivel de *cluster* y a nivel global de la red, definiendo un nivel de seguridad de red adecuado, en función del tipo de comunicación requerido.

  Teniendo en consideración la limitación, en términos de procesado y memoria, de los nodos IoT desplegados, se está trabajando en la adaptación de las técnicas y las claves previamente citadas, considerando las limitaciones asociadas a estos dispositivos. En este sentido, la particularidad de la provisión de dos interfaces radio por parte de los nodos desplegados, permite asignar diferentes niveles de seguridad a cada una de las mismas, dependiendo del grado de seguridad que se quiera asociar al servicio y gestión de la red, así como la posibilidad de incluir una librería específica que permita a los experimentadores utilizar alguna de estas técnicas criptográficas, para las comunicaciones asociadas a sus experimentos.
- Interacción entre nodos móviles y fijos; inclusión de nuevos nodos en la red: Como ya se describió de manera detallada en el Capítulo 3, tanto los nodos fijos como móviles ofrecen dos interfaces radio. En los primeros, la interfaz *Digimesh* proporciona el enrutamiento entre los nodos dentro de cada *cluster* para dar soporte a la provisión de servicio y a la gestión de la red; mientras que en la red móvil es la interfaz GPRS la que ofrece las capacidades de comunicación

asociadas a la provisión de servicio y a la gestión de la red. Adicionalmente, tanto los nodos móviles como los fijos están provistos de una interfaz 802.15.4 nativa que permite la comunicación entre ellos de manera directa, sin necesidad de tener que enviar la información a través de la plataforma central.

El uso de la interfaz 802.15.4 nativa permite la comunicación vehículo a vehículo (*V2V, vehicle to vehicle*), vehículo a nodo fijo (*V2I, vehicle to infrasructure*) [FIA13] [FUNEMS13] y nodo fijo a nodo fijo (*I2I, infrastrucrure to infrastructure*), siempre que los nodos se encuentren en zona de cobertura (la interfaz 802.15.4 no implementa ningún protocolo de enrutamiento), o a varios saltos si se implementa un protocolo de enrutamiento a nivel 802.15.4. Además, la capacidad de reprogramación remota sobre los nodos desplegados, los otorga de una amplia flexibilidad y capacidad de adaptación, permitiendo variar el comportamiento de los mismos acorde a la funcionalidad deseada.

- Mejora de la gestión de la plataforma: El protocolo de reprogramación remota de los nodos (MOTAP), definido e implementado en el Capítulo 4 de este trabajo, es un protocolo que se basa en el envío del código segmentado en fragmentos y el posterior reenvío, por parte del servidor y bajo demanda, de los paquetes perdidos por los nodos destino. Este modo de transmisión, se traduce en un funcionamiento ineficiente en entornos altamente interferentes, puesto que la aparición de desvanecimientos de larga duración (por ejemplo asociados a pérdidas de conexión nodo-servidor), puede acarrear una amplia pérdida de paquetes o la expiración del temporizador de espera, con la consiguiente detención del proceso de reprogramación. En este sentido, técnicas epidémicas, como las usadas en Deluge, o basadas en códigos LT (Luby transform), como los utilizados en SYNAPSE, permiten conseguir mejores rendimientos en este tipo de entornos interferentes. Actualmente, se está trabajando para mejorar el comportamiento del protocolo MOTAP actual en aras de conseguir, mediante diferentes técnicas de codificación de red, un rendimiento óptimo en entornos radio muy interferentes, como los que se pueden producir dentro de ciertos entornos urbanos.

Todas las líneas futuras previamente descritas, tanto aquellas inscritas dentro del marco de trabajo del proyecto SmartSantander (en su fase final), como las surgidas bajo el paraguas de otros proyectos europeos donde la plataforma de SmartSantander se presenta como habilitadora para experimentación y servicio comparten, como objetivo común, la mejora de la citada plataforma. La consecución de esta mejora se cimenta en la evolución desde diferentes planos de actuación, tales como, la generación de nuevos servicios de valor añadido en la plataforma, la interacción con otras fuentes de información (por ejemplo, redes sociales), el uso de técnicas de computación más avanzadas (basadas en la nube), la federación con otros despliegues para ofrecer un campo de pruebas para la experimentación más completo, la flexibilidad y la escalabilidad de la red para añadir fácilmente nuevos dispositivos y , por último, la mejora de la gestión de la red en términos de seguridad y acceso remoto a los nodos desplegados. La convergencia de los resultados derivados del trabajo en todas estas líneas de actuación, debe resultar en la consecución de una arquitectura para la gestión de una ciudad inteligente, lo suficientemente genérica, flexible, escalable y adaptable a las condiciones cambiantes asociadas a los entornos urbanos futuros, principalmente desde el punto de vista social (ciudadano), así como en lo referente a la sostenibilidad de los servicios públicos, aunque sin olvidar la componente experimental como campo de pruebas de servicios y aplicaciones futuras en entornos reales.

PUBLICACIONES

REVISTAS INTERNACIONALES

[Sooriyabandara08] Mahesh Sooriyabandara , Tim Farnham , Costas Efthymiou , Matthias Wellens , Janne Riihijärvi , Petri Mähönen , Alain Gefflaut , José Antonio Galache , Diego Melpignano , Arthur van Rooijen, "Unified Link Layer API: A generic and open API to manage wireless media access", Computer Communications, v.31 n.5, p.962-979, March, 2008

*Luis Sánchez, Luis Muñoz, José Antonio Galache, Pablo Sotres, Juan R. Santana, Verónica Gutiérrez, Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis, Dennis Pfisterer, "SmartSantander: IoT Experimentation over a Smart City Testbed", Computer Networks (**Pendiente de segunda revisión**).*

CONGRESOS INTERNACIONALES

[Galache07] José A. Galache, Verónica Gutiérrez, Ramón Agüero, Luis Muñoz, "Towards the integration of heterogeneous wireless sensor platforms: a generic API approach", Proceedings of the 2007 International Conference on Sensor Technologies and Applications, Valencia, Spain, Oct. 2007, pp. 411-417.

[Pentikousis07] Kostas Pentikousis, Ramón Agüero, Jens Gebert, José A. Galache, Oliver Blume, Pekka, Pääkkönen, "The Ambient Networks Heterogeneous Access Selection Architecture", Proc. First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management (M2NM), Sydney, Australia, October 2007, pp. 49-54

[Hernández11] José M. Hernández- Muñoz, Jesús Bernat, Luis Muñoz, Jose A. Galache, Mirko Presser, Luis A. Hernández, Jan Petterson, "Smart Cities at the Forefront of the Future Internet", J. Domingue et al. (Eds.): Future Internet Assembly, LNCS 6656, pp. 447–462, 2011.

[Sánchez11] Luis Sánchez, Jose A. Galache, Verónica Gutiérrez, Jose M. Hernández, Jesús Bernat, Alex Gluhak, Tomás García, "SmartSantander: The meeting point between Future Internet research and experimentation and the smart cities", Future Network & Mobile Summit (FutureNetw), 2011, Warsaw, Poland, June 2011.

[Galache12] Jose A. Galache, Juan R. Santana, Verónica Gutiérrez, Luis Sánchez, Pablo Sotres y Luis Muñoz, "Towards Experimentation-Service duality within a Smart City scenario", The 9th International Conference onWireless On-demand Network Systems and Services: WONS 2012, Courmayeur, Italia, Jan. 2012, pp. 175-181.

[Krčo13] Srdjan Krčo, Joao Fernandes, Luis Sanchez, Michele Natti, Evangelos Theodoridis, Divna Vučković, J. Casanueva, J.A. Galache, V. Gutiérrez, J.R. Santana, P. Sotres, "SmartSantander – a smart city experimental platform". Electrotechnical Review. 79 - 5, pp. 268 - 272. 07/01/2013. ISSN 2232-3228

[Galache13a] Jose Antonio Galache, Pablo Sotres, Juan Ramón Santana, Verónica Gutiérrez, Luis Sánchez and Luis Muñoz, "A Living Smart City: Dynamically Changing Nodes Behavior Through Over the Air Programming", International Workshop on Pervasive Internet of Things and Smart Cities (PITSaC 2013), Barcelona, Spain, March 25-28, 2013, pp. 1271-1276.

[Gutiérrez13] Verónica Gutiérrez, José A. Galache, Luis Sánchez, Luis Muñoz, Jose M. Hernández, Joao Fernandes, Mirko Presser, "SmartSantander: Internet of Things Research and Innovation through Citizen Participation", Galis, Alex; Gavras, Anastasius (Eds.): FIA 2013, LNCS 7858, pp. 173-186, 2013.

[Sánchez13] Luis Sánchez, Verónica Gutiérrez, José A. Galache, Pablo Sotres, Juan Ramón Santana, Javier Casanueva and Luis Muñoz, "SmartSantander: Experimentation and Service Provision in the Smart City", Global Wireless Summit 2013, Atlantic City, New Jersey, USA, June 2013.

[Galache13b] José A. Galache, Verónica Gutiérrez, Juan R. Santana, Luis Sánchez, Pablo Sotres, Javier Casanueva, Luis Muñoz, "SmartSantander: A joint service provision facility and experimentation-oriented testbed, within a smart city environment", Future Network & Mobile Summit 2013, Lisbon, Portugal, July, 2013.

CONGRESOS NACIONALES

[Galache08] José A. Galache, Ramón Agüero, Johnny Choque, Luis Muñoz, "Plataforma para la gestión y monitorización de múltiples interfaces heterogéneas subyacentes", VII Jornadas de Ingeniería Telemática, Alcalá de Henares, Madrid, Sept. 2008.

[Galache11] José A. Galache, Verónica Gutiérrez, Luis Sánchez, Juan R. Santana, Luis Muñoz, David Gascón, José M. Hernández, Beatriz Sarmiento. Laura González, "Experimentación en la Internet del Futuro sobre una Red de Sensores para la Gestión de Aparcamiento en una Ciudad Inteligente", XXI Jornadas Telecom I+D, Santander, España, Sept. 2011.

LISTA DE ACRÓNIMOS

AES	Advanced Encryption Standard
AES128	Advanced Encryption Standard 128
AKARI	Architecture Design Project for New Generation Network
AMPLIFIRE	Amplifying Future Internet Research and Experimentation for a Sustainable Future
AN	Ambient Networks
ANA	Autonomic Network Architecture
ANP2	Ambient Networks Phase 2
AP	Access Point
API	Application Programming Interface
ARQ	Automatic Repeat Request
ASCII	American Standard Code for Information Interchange
BBDD	Base de Datos
CAN	Controller Area Network
CAN-Bus	Controller Area Network vehicle bus standard
CELTIC	Cooperation for a European sustained Leadership in Telecommunications
CERNET	China Education and Research Network
CitySDK	City Service Development Kit
ClouT	Cloud of Things for empowering the citizen clout in smart cities
CO	Monóxido de Carbono
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
Ctrl1…n	Controlador 1…n
DCA	Data Collection and Analisys
DSSS	Direct Sequence Spread Spectrum
EDGE	Enhanced Data rates for Global Evolution
EGPRS	Enhanced GPRS
EIFFEL	Evolved Internet Future for European Leadership
EMFs	ElectroMagnetic Fields
ENOLL	European Network of Living Labs
FED4FIRE	Federation for Future Internet Research and Experimentation
FEDERICA	Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures
FI	Future Internet
FIA	Future Internet Assembly
FIBRE	Future Internet testbeds/experimentation between BRazil and Europe
FI-LAB	FI-WARE Open Innovation Lab
FI-PPP	Future Internet Public-Private Partnership Programme
FIRE	Future Internet Research and Experimentation
FIRESTATION	FIRE Support Action
FP1	Primer Programa Marco
FP7	Séptimo Programa Marco
GENI	Global Environment for Network Innovations
GIT	Grupo de Ingeniería Telemática
GLL	Generic Link Layer

gll_AL	GLL Abstraction Layer
gll_IM	GLL Interface & Management
GOLLUM	Generic Open Link-Layer API for Unified Media Access
GPRS	General Packet Radio Service
GPS	Global Positioning System
GW-EXP	Servidor Central y la Pasarela de Experimentación
hARQ	Hybrid Automatic Repeat Request
HSPA	High Speed Packet Access
HW	Hardware
I+D	Investigación y desarrollo
I+D+i	Investigación , desarrollo e innovación
I2C	Inter-Integrated Circuit
I2I	infrastructure to infrastructure
ICA	interfaz de control de acceso
ICT	Information and Communication Technologies
IDAS	Intelligence Data Advanced Solution
IEEE	Institute of Electrical and Electronics Engineers
IERC	IoT European Research Cluster
IMT	International Mobile Telecommunications
IoS	Internet of Services
IOS	iPhone Operating System
IoT	Internet of Things
IoT-A	Internet of Things - Architecture
IP	Internet Protocol
IR	Infrarrojo
ISA	interfaz de soporte a las aplicaciones
ISE	interfaz de soporte a la experimentación
ISG	interfaz de soporte a la gestión
ISM	Industrial, Scientific, Medical
IST-FP6	Information Society Technologies – 6th Framework Programme
ITEA	Information Technology for European Advancement
ITU	International Telecommunications Union
iWSN	API para la gestión de la Red de Sensores
KPIs	Key Performance Indicators
LED	Light-Emitting Diode
Lexnet	Low EMF Exposure Networks
LL	Living Labs
LLA	Link Layer Adapter
LT	Luby transform
LTE	Long Term Evolution
M2M	Machine to Machine
MAC	Medium Access Control
MAGNET	My Personal Adaptive Global NET
MIMO	Multiple Input Multiple Output
MM	Mac Mode
MNP	Multihop Network Programming

MOAP	Multihop Over-the-Air Programming
MOBILIA	Mobility concepts for IMT-Advances
MOTAP	Multihop OTAP
MR	Mesh Network Retries
MT	Multiple Transmissions
NFC	Near Field Communication
NH	Network Hops
NN	Network Delay Slots
NO2	Dióxido de Nitrógeno
NOVI	Networking innovations Over Virtualized Infrastructures
NSF	National Science Foundation
NSF	National Science Foundation
O&M	observaciones y medidas
O3	Ozono
OFDM	Orthogonal Frequency Division Multiplexing
OFELIA	OpenFlow in Europe: Linking Infrastructure and Applications
OLA	Ordenanza Limitadora de Aparcamiento
OMS	Organización Mundial de la Salud
ORBIT	Open-Access Research Testbed for Next-Generation Wireless Networks
OTAP	Over The Air Programing
PAN	Personal Area Networks
PANID	Personal Area Network Identifier
Panlabll	Pan European Laboratory Infrastructure Implementation
PAS_MOV	Pasarela para Nodos Móviles
PAS_SER	Gestor de los Nodos ubicado en la Pasarela para Servicio
PCTCAN	Parque Científico y Tecnológico de Cantabria
PECES	PErvasive Computing in Embedded Systems
PIR	Passive Infrared
PLAMIN	Plataforma de Adaptación de Múltiples Interfaces
PlanetLab	An open platform for developing, deploying, and accesssing planetary-scale services
PM10	Partículas Materiales de tamaño inferior o igual a 10 micras
PoE	Power over Ethernet
PROFIT	Programa de Fomento de la Investigación Técnica
PWM	Pulse Width Modulation
QoS	Quality of Service
QR	Quick Response
RADICAL	RApid Deployment and adoption of sustainable socially-aware and intelligent sensing services for emerging smart cities
RAM	Random-access memory
RAT	Radio Access Technology
REST	Representational State Transfer
RFID	Radio frequency identification
RISC	Reduced Description Set Computer
RR	MAC Retries
RS	Reservation System
RS	Reservation System

RSSI	Received Signal Strength Indicator
RTT	Round-Trip delay Time
SD	Secure Digital
SDRAM	Synchronous dynamic random access memory
SENSEI	Integrating the Physical with the Digital World of the Network of the Future
SensorGrid4Env	Semantic Sensor Grids for Rapid Application Development for Environmental Management
SGR	subsistema de gestión de la red
SMARTiP	Smart Metropolitan Areas Realised Through Innovation & People
SNAA	Sensor Network Authentication and Authorization
SPITFIRE	Semantic Service Provisioning for the Internet of Things using Future Internet Research by Experimentation
SQL	Structured query language
SR	Sistema de Reservas
SReluge	Secure Rateless Deluge
SSA	subsistema de soporte a las aplicaciones
SSA	subsistema de autenticación y autorización
SSE	subsistema de soporte a la experimentación
SSID	Service Set Identifier
TCP/IP	Transmission Control Protocol/Internet Protocol
TDMA	Time Division Multiple Access
TEFIS	Testbed for Future Internet Services
TIC	Tecnologías de la Información y la Comunicación
TR	Testbed Runtime
TTL	Time To Live
UART	Universal Asynchronous Receiver/Transmitter
UCL	Universal Convergence Layer
UCP	ULLA Command Processing
UEP	ULLA Event Processing
ULLA	Unified Link Layer API
ULP	Unidad local de proceso (CLV)
UMTS	Universal Mobile Telecommunications System
UQL	ULLA Query Language
UQP	ULLA Query Processing
USB	Universal Serial Bus
USN	Ubiquitous Sensor Network
V2I	vehicle to infrastructure
V2V	vehicle to vehicle
WAVE	Wireless Access in Vehicular Environments
WECA	Wireless Ethernet Compatibility Alliance
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WISEBED	Wireless Sensor Network Testbeds
WSNs	Wireless Sensor Networks
XML	eXtensible Markup Language

REFERENCIAS

[Schaffers11] Schaffers, H.; Komninos, N.; Pallot, M.; Trousse, B.; Nilsson, M.; Oliveira, A.; "Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation". In Future Internet Assembly, LNCS 6656, Publisher: Springer Berlin (Heidelberg), 2011; pp. 431–446.

[Nam11] Nam, T.; Pardo, T. A.; "Conceptualizing smart city with dimensions of technology, people, and institutions". In Proceedings of the 12th Annual International Digital Government Research Conference, College Park (Maryland), USA; 12-15 June 2011; pp. 282-291.

[Hernández11] José M. Hernández- Muñoz, Jesús Bernat, Luis Muñoz, Jose A. Galache, Mirko Presser, Luis A. Hernández, Jan Petterson, "Smart Cities at the Forefront of the Future Internet", J. Domingue et al. (Eds.): Future Internet Assembly, LNCS 6656, pp. 447–462, 2011.

[Pan11] J. Pan, P. Subharthi, and R. Jain, "A survey of the research on future internet architectures", IEEE Communications Magazine, vol. 49, no. 7, pp. 26-36, July 2011.

[M2M] OECD (2012), "Machine-to-Machine Communications: Connecting Billions of Devices", OECD Digital Economy Papers, No. 192, OECD Publishing.

[Botterman09] M. Botterman, "Internet of Things: an early reality of the Future Internet", Workshop Report prepared for European Commission, Information Society and Media Directorate General, Networked Enterprise & RFID Unit (D4), Prague, May 2009.

[Ashton09] Kevin Ashton, first person in using Internet of Things as title of a presentation, http://kevinjashton.com/2009/06/22/the-internet-of-things/

[Akyildiz02] Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E.; "Wireless sensor networks: a survey," Computer Networks, vol. 38, no 4, March 2002; pp. 393-422.

[FIRE] Iniciativa FIRE, http://cordis.europa.eu/fp7/ict/fire/

[Gluhak11] Gluhak, A., Krco, S., Nati, M., Pfisterer, D., Mitton, N., Razafindralambo, T., "A survey on facilities for experimental internet of things research", Communications Magazine, IEEE , vol.49, no.11, pp.58-67, November 2011

[ENOLL] Red europea de laboratorios vivos (European Network of Living Labs), http://www.openlivinglabs.eu/

[Want06] R. Want, "An Introduction to RFID Technology", IEEE Pervasive Computing, 5(1):25-33, Jan.-March 2006

[IEEE802.15.4] Estándar IEEE 802.15.4, http://www.ieee802.org/15/pub/TG4.html

[Craig03] William C. Craig "ZigBee: Wireless Control That Simply Works", ZigBee Alliance, 2003

[Digimesh] Protocolo DIgimesh, http://www.digi.com/technology/digimesh/

[García-Hernando08] García-Hernando, A.-B.; Martínez-Ortega, J.-F.; López-Navarro, J.-M.; Prayati, A.; Redondo-López, L. (Eds.), "Problem Solving for Wireless Sensor Networks", Springer Publishing Company, Incorporated, 2008.

[CORONIS] Plataforma de comunicación para redes de sensores, http://www.coronis.com/

[IEEE802.15.1] Estándar IEEE 802.15.1-2002, http://www.ieee802.org/15/pub/TG1.html

[Ferro05] E. Ferro and F. Potorti, "Bluetooth and Wi-Fi wireless protocols: A survey and a comparison," IEEE Wireless Commun., vol. 12, no. 1, pp. 12-16, Feb. 2005.

[Shrivastava08] V. Shrivastava, S. Rayanch, J. Yoon, and S. Banerjee, "802.11n Under the Microscope",in IMC, October 2008

[Jiang08] D. Jiang and L. Delgrossi, "Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments," in Vehicular Technology Conference, 2008. IEEE, 2008, pp. 2036-2040.

[Uzcategui09] R. Uzcategui and G. Acosta-Marum, "WAVE: a tutorial," Communications Magazine, IEEE, vol. 47, no. 5, pp. 126-133, 2009.

[Ghribi00] Brahim Ghribi, Luigi Logrippo, "Understanding GPRS: the GSM packet radio service, Computer Networks", The International Journal of Computer and Telecommunications Networking, v.34 n.5, p.763-779, Nov. 2000.

[Richardson00] K.W. Richardson, "UMTS overview," Electronics & Communication Engineering Journal, 12(3): pages 93-100, 2000.

[Cox12]Christopher Cox, "An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications", John Wiley & Sons Ltd, 2012.

[LIBELIUM] Compañía especializada en redes de sensores, http://www.libelium.com/

[DIGI] Fabricante de módulos radio para redes de sensores, http://www.digi.com/

[MEMSIC] Dispositivos embebidos (anteriormente CROSSBOW), http://www.memsic.com

[URBIOTICA] Empresa especializada en redes de sensores, http://www.urbiotica.com

[ADVANTIC] Empresa especializada en redes de sensores y sistemas de monitorización remota, http://www.advanticsys.com

[SENTILLA] Empresa especializada en redes de sensores, http://www.sentilla.com

[TST] Empresa especializada en redes de sensores, http://www.tst-sistemas.es

[ARDUINO] Plataforma de prototipado hardware/software, http://www.arduino.cc/

[Wiring] Marco de programación de código abierto para microcontroladores, http://wiring.org.co/

[Processing] Lenguaje de programación y entorno de desarrollo, http://processing.org/

[Dutta05] A. Dutta et al, "Seamless Handover across Heterogeneous Networks – An IEEE802.21 Centric Approach", WPMC 2005.

[Sooriyabandara06] M. Sooriyabandara, T. Farnham, C. Efthymiou, M.Wellens,K. Rerkrai, M. Bandholz, P. Mähönen, J. Riihijärvi, D. Melpignano, D.Siorpaes, A. Gefflaut, V. Gutiérrez Unified Link Layer API: Design and Initial Implementation ResultsProceedings of IST Mobile Summit 2006, Mykonos, Greece, June 2006

[Belgasmi08] F. Belqasmi, R. Glitho, and R. Dssouli, "Ambient Network Composition," IEEE Trans. Network, vol. 22, pp. 6-12, July 2008.

[Prasad10] Prasad, Ramjee (Ed.), "My personal Adaptive Global NET (MAGNET)", Signals and Communication Technology, Springer 2010.

[mCiudad] Proyecto científico-tecnológico singular y de carácter estratégico para el Fomento de las Plataformas Tecnológicas Españolas m:Ciudad.

[PROFIT] Programa nacional PROFIT, http://www.minetur.gob.es/portalayudas/profit/

[EASY_WIRELESS] Proyecto ITEA Easy Wireless, http://ew.thales.no

[ITEA] Programa ITEA, http://www.itea2.org/

[MOBILIA] Proyecto CELTIC MOBILIA, http://www.mobilia-project.org/

[CELTIC] Programa Celtic: Hacia un mundo conectado inteligente, http://www.celticplus.eu/

[IEEE802.16] IEEE 802.16 Working Group on Broadband Wireless Access Standards, http://www.ieee802.org/16/

[Galache08] José A. Galache, Ramón Agüero, Johnny Choque, Luis Muñoz", Plataforma para la gestión y monitorización de múltiples interfaces heterogéneas subyacentes", VII Jornadas de Ingeniería Telemática, Alcalá de Henares, Madrid, Sept. 2008.

[Sooriyabandara08] Mahesh Sooriyabandara , Tim Farnham , Costas Efthymiou , Matthias Wellens , Janne Riihijärvi , Petri Mähönen , Alain Gefflaut , José Antonio Galache , Diego Melpignano , Arthur van Rooijen, "Unified Link Layer API: A generic and open API to manage wireless media access", Computer Communications, v.31 n.5, p.962-979, March, 2008

[Galache07] José A. Galache, Verónica Gutiérrez, Ramón Agüero, Luis Muñoz, "Towards the integration of heterogeneous wireless sensor platforms: a generic API approach", Proceedings of the 2007 International Conference on Sensor Technologies and Applications, Valencia, Spain, Oct. 2007, pp. 411-417.

[Wettern97] Joern Wettern, Karanjit S. Siyan, "Inside TCP/IP" New Riders Publishing; 3 Sub edition.

[Agüero07] R. Agüero, J. Gebert, J. Choque and H. Eckhardt, "Towards a Multi-Access Prototype in Ambient Networks", IEEE 16th IST Mobile and Wireless Communications Summit, Budapest, Hungary, June 2007, pp.1 -5.

[Pentikousis07] Kostas Pentikousis, Ramón Agüero, Jens Gebert, José A. Galache, Oliver Blume, Pekka, Pääkkönen, "The Ambient Networks Heterogeneous Access Selection Architecture", Proc. First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management (M2NM), Sydney, Australia, October 2007, pp. 49-54

[Micaz] Módulo Micaz, http://www.memsic.com/wireless-sensor-networks/MPR2400CB

[Mica2] Módulo Mica2 obsoleto y sustituido por el módulo Cricket a 433 MHz, http://www.memsic.com/wireless-sensor-networks/MCS410CA

[FIA] Asamblea de la Internet del futuro, http://www.future-internet.eu/home/future-internet-assembly.html.

[FP7] 7° Programa Marco para la Investigación y el Desarrollo Tecnológico (2007-2013), http://cordis.europa.eu/fp7

[Gavras07] Anastasius Gavras , Arto Karila , Serge Fdida , Martin May , Martin Potts, Future internet research and experimentation: the FIRE initiative, ACM SIGCOMM Computer Communication Review, v.37 n.3, July 2007

[Mähönen06] Petri Mähönen et al., " EIFFEL: Evolved Internet Future for European Leadership", white paper from the EIFFEL ThinkTank, Dec 2006, http://www.fp7-eiffel.eu/fileadmin/docs/EIFFEL-FINAL.pdf

[FI-WARE] Plataforma núcleo de la Internet del futuro, http://www.fi-ware.eu/

[FI-PPP] Programa de colaboración público-privada en el marco de la Internet del Futuro, http://www.fi-ppp.eu

[FIRESTATION] Acción de soporte para FIRE, http://www.ict-fire.eu/home/firestation.html.

[AMPLIFIRE] Amplificando FIRE para un future sostenible (continuación de FIRESTATION), http://www.ict-fire.eu/home/amplifire-project.html

[Komninos11] Nicos Komninos, Hans Schaffers, Marc Pallot, "Developing a Policy Roadmap for Smart Cities and the Future Internet", In Proceedings of the eChallenges 2011 Conference, 24-26th October 2011, Florence.

[ENOLL] Red europea de laboratorios vivos, http://www.openlivinglabs.eu/

[OneLab] Bancos de prueba para la Internete del futuro, http://www.onelab.eu/

[Tranoris12] Christos Tranoris, and Spyros G. Denazis, "OpenFlow and P2P Integrated Testing, Project: OpenLab", TRIDENTCOM, volume 44 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, page 377-378. Springer, (2012)

[Lymberopoulos12] L. Lymberopoulos, M. Grammatikou, M. Potts, P. Grosso, A. Fekete, B. Belter, M. Campanella and V. Maglaris, "NOVI Tools and Algorithms for Federating Virtualized

Infrastructures," to appear in Future Internet – From Technological Promises to Reality, Springer Lecture Notes in Computer Science, pp. 213-224, 2012.

[F-LAB]Proyecto F-Lab: Federando recursos computacionales, http://f-lab.fr/

[Sallent12] S. Sallent, A. Abelem, I. Machado., et al.. FIBRE project: Brazil and Europe unite forces and testbeds for the Internet of the future. In: TridentCom 2012, Thessaloniki, Greece, June 2012. Procedings of TridentCom 2012, 2012.

[G-LAB] Facilidad experimental y de investigación alemana, http://www.german-lab.de/

[Szegedi10] P.Szegedi, J.Ferrer Riera, J.A.García-Espín, M.Hidell, P.Sjödin, P.Söderman, M.Ruffini, D.O'Mahony,A.Bianco, L.Giraudo, M.Ponce de Leon, G.Power, C.Cervelló-Pastor, V. López, S.Naegele-Jackson, "Enabling Future Internet Research: The FEDERICA Case", submitted to IEEE Communication magazine, 2010.

[Tschudin07] C. Tschudin, C. Jelger, "An "Autonomic Network Architecture" Research Project", In proceedings of Praxis der Informationsverarbeitung und Kommunikation (PIK Magazine), vol. 30, no. 1, pp. 26-31, January-March 2007.

[IERC] Grupo de investigación europeo de la Internet de las cosas, http://www.internet-of-things-research.eu

[HOR2020] Programa Marco de investigación para el período 2014-2020, http://ec.europa.eu/research/horizon2020

[ICT-FP7] Programa ICT (Information and Communication Technologies) dentro del FP7, http://cordis.europa.eu/fp7/ict/

[Coulson12] G. Coulson, et. al., "Flexible experimentation in wireless sensor networks", In Commun. ACM, ACM, volume 55, 2012.

[Kopsel11] A. Kopsel, ,H. Woesner, "OFELIA – Pan-European Test Facility for OpenFlow Experimentation", in Proceeding ServiceWave'11 Proceedings of the 4th European conference on Towards a service-based internet pp 311-312, 26 - 28 October 2011; Poznan, Poland

[Wahle11] S. Wahle, C. Tranoris, S. Denazis, A. Gavras, K. Koutsopoulos, T. Magedanz, and S. Tompros, "Emerging Testing Trends and the Panlab Enabling Infrastructure,"IEEE Communications Magazine, vol. 49, no. 3, pp. 167–175, March 2011, ISSN:0163-6804.

[FED4FIRE] Federation de diferentes infraestructuras FIRE para contruir una red a nivel europeo, http://fed4fire.eu/

[Selvarajah10] K. Selvarajah and N. Speirs, "Integrating smart spaces into the pervasive computing in embedded systems (PECES) project", IEEE CCNC, USA, pp. 1-2, Jan 9-12, 2010.

[Hume12] A. Hume, "BonFIRE: A Multi-cloud Test Facility for Internet of Services Experimentation", TRIDENTCOM, volume 44 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, page 81-96. Springer, (2012)

[IoT-A] Arquitectura de la Internet de las cosas, http://www.iot-a.eu

[OUTSMART] Michelle B. Mikkelsen et al., "OUTSMART ANNUAL REPORT 2013"

[SensorGrid4Env] Proyecto SensorGrid4Env: las TIC para un crecimiento sostenible, http://www.semsorgrid4env.eu/

[SMARTiP] SMARTiP Ciudadanos inteligentes en ciudades inteligentes, http://www.smart-ip.eu

[RADICAL] Rápido despliegue y adopción de nuevos servicios de medidas inteligentes, sostenibles y socialmente aceptados, para las ciudades inteligentes emergentes, http://www.radical-project.eu/

[PEOPLE] PEOPLE: Ciudades inteligentes para innovación inteligente, http://www.people-project.eu

[Sánchez11] Luis Sánchez, Jose A. Galache, Verónica Gutiérrez, Jose M. Hernández, Jesús Bernat, Alex Gluhak, Tomás García, "SmartSantander: The meeting point between Future Internet research and experimentation and the smart cities", Future Network & Mobile Summit (FutureNetw), 2011, Warsaw, Poland, June 2011, pp. 1-8.

[Krčo13] Srdjan Krčo, Joao Fernandes, Luis Sanchez, Michele Natti, Evangelos Theodoridis, Divna Vučković, J. Casanueva, J.A. Galache, V. Gutiérrez, J.R. Santana, P. Sotres, "SmartSantander – a smart city experimental platform". Electrotechnical Review. 79 - 5, Jan. 2013, pp. 268 – 272.

[TEFIS] TEFIS: Banco de pruebas para los servicios de la Internet del Futuro, http://www.tefisproject.eu/

[Niebert08] N. Niebert, S. Baucke, I. El-Khayat, M. Johnsson, B. Ohlman, H. Abramowicz, K. Wuenstel, H. Woesner, J. Quittek, and L. Correia, "The Way 4WARD to the Creation of a Future Internet," in Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on, Sep. 2008, pp. 1-5.

[Presser09] M. Presser, P.M. Barnaghi, M. Eurich, and C. Villalonga, "The SENSEI project: integrating the physical world with the digital world of the network of the future," Communications Magazine, IEEE, vol. 47, no. 4, pp. 1-4, Apr. 2009.

[CITYSDK] CitySDK: Desarrollo de servicios en la ciudad, http://www.citysdk.eu/

[Pfisterer11] D. Pfisterer, K. Romer, et al., "SPITFIRE: toward a semantic web of things," Communications Magazine, IEEE, Vol. 49, Issue 11, pp.40-48, Nov. 2011.

[ESINTERNET] Plataforma Tecnológica Española de Convergencia hacia Internet del Futuro, http://esinternet.imasdtic.es

[COL_EUR] Convenios de colaboración Científico/Técnica entre Europa y diversos países del mundo, http://ec.europa.eu/research/iscp/index.cfm?pg=countries

[GENI] Entorno global para innovaciones en la red, http://www.geni.net/.

[NSF] Agencia independiente del gobierno de Estados Unidos responsable de promover la ciencia y la tecnología, a través de programas de investigación y proyectos de educación, http://www.nsf.gov/.

[ORBIT] Banco de pruebas de acceso libre para la investigación en redes inalámbricas de próxima generación, http://www.orbit-lab.org/

[AKARI] Arquitectura AKARI, http://akari-project.nict.go.jp/eng/index2.htm

[CERNET] Infraestructura CERNET, http://www.cernet.edu.cn/

[KOREN] Iniciativa KOREN, http://www.koren21.net/

[PlanetLab] Una plataforma abierta para el desarrollo, despliegue y acceso a servicios a escala mundial, http://www.planet-lab.org/

[RED_SDR] Despliegue de dispositivos en Santander, http://maps.smartsantander.eu/

[Gutiérrez13] Verónica Gutiérrez, José A. Galache, Luis Sánchez, Luis Muñoz, Jose M. Hernández, Joao Fernandes, Mirko Presser, "SmartSantander: Internet of Things Research and Innovation through Citizen Participation", Galis, Alex; Gavras, Anastasius (Eds.): FIA 2013, LNCS 7858, pp. 173-186, 2013.

[Android] Sistema operativo para dispositivos móviles, http://www.android.com/

[IOS] Sistema operativo para dispositivos móviles, http://www.apple.com/es/ios/

[DENIPA] Ingeniería de Radiocomunicaciones y de instrumentación. http://www.denipa.es/

[FAGOR] Sistemas de gestión y localización de flotas a través de internet e intranet. http://www.fagorelectronica.es/flotas/indexflotas.html

[CAN Std.] Estándar de comunicación del bus CAN, ISO 11898

[Sánchez13] Luis Sánchez, Verónica Gutiérrez, José A. Galache, Pablo Sotres, Juan Ramón Santana, Javier Casanueva and Luis Muñoz, "SmartSantander: Experimentation and Service Provision in the Smart City", Global Wireless Summit 2013, Atlantic City, New Jersey, USA, June 2013.

[Galache13b] José A. Galache, Verónica Gutiérrez, Juan R. Santana, Luis Sánchez, Pablo Sotres, Javier Casanueva, Luis Muñoz, "SmartSantander: A joint service provision facility and experimentation-oriented testbed, within a smart city environment", Future Network & Mobile Summit 2013, Lisbon, Portugal, July, 2013.

[Telco] Bernat, J.; Pérez, S.; González, A.; Sorribas, R.; Villarrubia, L.; Hernández, L.,"Ubiquitous Sensor Networks in IMS: an Ambient Intelligence Telco Platform," in ICT Mobile Summit, Stokholm, Sweeden, 2008.

[Shibboleth] Shibboleth: Proyecto de código abierto que prove capacidades para autenticación, http://shibboleth.net/.

[ProtoBuf] Fornato de intercambio de datos de Google, http://code.google.com/p/protobuf/.

[DCA_IDAS] Plataforma de Telefónica para la gestión de ciudades inteligentes, https://m2m.telefonica.com/

[GE_DCA] DCA IDAS: Principal componente del habilitador genérico 'Back-end Device management GE', http://www.fi-ware.eu/2013/05/14/fi-ware-opens-smartcities-to-future-internet-app-developers/

[OPEN_DATA_SITES] Resumen de iniciativas de "open data" a nivel mundial, http://www.data.gov/opendatasites

[Galache13a] Jose Antonio Galache, Pablo Sotres, Juan Ramón Santana, Verónica Gutiérrez, Luis Sánchez and Luis Muñoz, "A Living Smart City: Dynamically Changing Nodes Behavior Through Over the Air Programming", International Workshop on Pervasive Internet of Things and Smart Cities (PITSaC 2013), Barcelona, Spain, March 25-28, 2013, pp. 1271-1276.

[Brown06] Brown, S., Sreenan, C.J.: Updating Software in Wireless Sensor Networks: A Survey. Technical Report UCC-CS-2006-13-07, University College Cork, Ireland (July 2006)

[Wang 06] Q. Wang, Y. Zhu, and L. Cheng. Reprogramming wireless sensor networks: Challenges and approaches. IEEE Network Magazine, 20(3):48- 55, May-June 2006.

[Jeong03] J. Jeong, S. Kim, and A. Broad, "Network Reprogramming", Berkeley, California, USA, Aug. 2003. [Online]. Available: http://www.tinyos.net/tinyos-1.x/doc/

[TinyOS] Sistema operativo diseñado para dispositivos de bajo consumo, http://www.tinyos.net/

[Stathopoulos03] T. Stathopoulos, J. Heidemann, and D. Estrin. "A remote code update mechanism for wireless sensor networks". Technical report, UCLA, 2003.

[Kulkarni04] Kulkarni, S.S., Arumugam, M.: Infuse: A TDMA Based Data Dissemination Protocol for Sensor Networks. Technical Report MSU-CSE-04-46. Dept. of Computer Science and Engineering, Michigan State University, MI (2004).

[Beutel04] Beutel, J., Dyer, M., Meier, L., Ringwald, M., Thiele, L.: Next-Generation Deployment Support for Sensor Networks. TIK-Report No: 207. Computer Engineering and Networks Lab, Swiss Federal Institute of Technology (ETH), Zurich (2004).

[Kulkami05] S. S. Kulkarni and L. Wang, "MNP: Multihop Network Reprogramming Service for Sensor Networks," in IEEE ICDCS, Columbus, Ohio, USA, Jun. 2005.

[Naik05] V. Naik et al., "Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Wireless Embedded Devices," 26th IEEE Real-Time Sys. Symp. Dec. 2005.

[Levis04] P. Levis and D. Culler, "The Firecracker Protocol," Proc. 11th ACM SIGOPS Euro. Wksp., Leuven, Belgium, Sept. 2004.

[Rossi08] Rossi, M., Zanca, G., Stabellini, L., Crepaldi, R., Harris, A., Zorzi, M.: SYNAPSE: A network reprogramming protocol for wireless sensor networks using fountain codes. In: Proc. of SECON, San Francisco, CA (2008) 188{196}

[Hui04] J. W. Hui and D. Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale," in ACM SenSys, Baltimore, Maryland, USA, Nov. 2004.

[Luby97] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical lossresilient codes," in Prc. ACM Symp. Theory Comp., El Paso, TX, 1997, pp. 150–159.

[Luby01] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 569–584, 2001.

[Hagedorn08] Andrew Hagedorn , David Starobinski , Ari Trachtenberg, Rateless Deluge: Over-the-Air Programming of Wireless Sensor Networks Using Random Linear Codes, Proceedings of the 7th international conference on Information processing in sensor networks, p.457-466, April 22-24, 2008.

[Law11] Law, Yee Wei and Zhang, Yu and Jin, Jiong and Palaniswami, Marimuthu and Havinga, Paul (2010) Secure Rateless Deluge: Pollution-Resistant Reprogramming and Data Dissemination for Wireless Sensor Networks. Journal on Wireless Communications and Networking, 2011 . p. 685219. ISSN 1687-1472.

[Reijers03] Niels Reijers and Koen Langendoen. Efficient Code Distribution in Wireless Sensor Networks. In Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, pages 60{67. ACM Press, 2003.

[Jeong04] Jeong, J., D. "Incremental Network Programming for Network Sensors", In: Proc. of the 1 st Annual IEEE Communications Society Conf. on Sensor and Ad Hoc Networks and Communications (SECON 2004). IEEE(2004) 25-33

[Koshy05] Koshy, J., Pandy, R.: "Remote Incremental Linking for Energy-Efficient Reprogramming of Sensor Networks". In: Proc. of the 2nd European Workshop on Wireless Sensor Networks (EWSN'05). IEEE (2005) 354-365

[Galache12] Jose A. Galache, Juan R. Santana, Verónica Gutiérrez, Luis Sánchez, Pablo Sotres y Luis Muñoz, "Towards Experimentation-Service duality within a Smart City scenario", The 9th International Conference onWireless On-demand Network Systems and Services: WONS 2012, Courmayeur, Italia, Jan. 2012, pp. 175-181.

[Galache11] José A. Galache, Verónica Gutiérrez, Luis Sánchez, Juan R. Santana, Luis Muñoz, David Gascón, José M. Hernández, Beatriz Sarmiento. Laura González, "Experimentación en la Internet del Futuro sobre una Red de Sensores para la Gestión de Aparcamiento en una Ciudad Inteligente", XXI Jornadas Telecom I+D, Santander, España, Sept. 2011.

[Markoff08] J. Markoff ,"Can't Find a Parking Spot? Check Smartphone", New York Times, 12 July 2008.

[NEDAP] Solución de NEDAP para la gestión de plazas de aparcamiento http://www.nedap.es/pdf/SENSIT2.pdf

[Urbiotica] Solución de Urbiotica para la gestión de plazas de aparcamiento, http://www.urbiotica.com/productos/u-spot/

[Tinynode] Solución de Tinynode para la gestión de plazas de aparcamiento, http://www.tinynode.com/?q=node/117

[Smartgrains] Solución de Smartgrains para la gestión de plazas de aparcamiento, http://www.smartgrains.com/en/parking-innovation/#sensors

[Libelium] Solución de Libelium para la gestión de plazas de aparcamiento, http://www.libelium.com/development/waspmote/documentation/smart-parking-board-technical-guide/

[SFPark] Proyecto desarrollado en la ciudad de San Francisco para la gestión integral de las plazas de aparcamiento, http://sfpark.org/

[Streetline] Solución de Streetline para la gestión de plazas de aparcamiento, http://www.streetline.com/

[Fybr] Solución de Fybr para la gestión de plazas de aparcamiento, http://www.fybr-tech.com/how-fybr-works/our-system/

[WorldSensing] Solución de WorldSensing para la gestión de plazas de aparcamiento, http://www.worldsensing.com/smart-industries/smart-cities.html

[ESMARTCITY] Información sobre ciudades inteligentes, http://www.esmartcity.es

[Hollosi13] Danilo Hollosi et al., "Enhancing Wireless Sensor Networks with Acoustic Sensing Technology: Use Cases, Applications & Experiments", International Conference on Internet of Things (iThings2013), Beijing, China, 2013.

[ClouT] Proyecto de colaboración Unión Europea-Japón, http://clout-project.eu/

[Lexnet] Lexnet: Redes de baja exposición a campos electromagnéticos, http://www.lexnet-project.eu/

[OMS11] Campos electromagnéticos y salud pública, teléfonos móviles, http://www.who.int/mediacentre/factsheets/fs193/en/index.html

[AES128] Descripción del protocolo de encriptación AES128, RFC3826

[FIA13] Future Internet Assembly 2013, Dublin, Ireland. http://www.fi-dublin.eu/

[FUNEMS13] Future Networks & Mobile Summit 2013, http://www.futurenetworksummit.eu/2013/