

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN



TRABAJO FIN DE GRADO

Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

ADAPTACIÓN Y REESTRUCTURACIÓN DE LA
IMPLEMENTACIÓN DE UNA ARQUITECTURA
COGNITIVA PARA REDES DE SENsoRES
INALÁMBRICAS

Jose María Bermudo Mera

Julio 2014

TRABAJO FIN DE GRADO

Título: ADAPTACIÓN Y REESTRUCTURACIÓN DE LA IMPLEMENTACIÓN DE UNA ARQUITECTURA COGNITIVA PARA REDES DE SENSORES INALÁMBRICAS

Autor: JOSE MARÍA BERMUDO MERA

Tutor: ELENA ROMERO PERALES

Ponente: ALVARO ARAUJO PINTO

Departamento: DEPARTAMENTO DE INGENIERÍA ELETRÓNICA

TRIBUNAL

Presidente: D. OCTAVIO NIETO-TALADRIZ GARCÍA
Vocal: D. FERNANDO GONZÁLEZ SANZ
Secretario: D. ALVARO ARAUJO PINTO
Vocal suplente: D. PEDRO JOSÉ MALAGÓN MARZO

Calificación:

Madrid, a de de

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN



TRABAJO FIN DE GRADO

Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

ADAPTACIÓN Y REESTRUCTURACIÓN DE LA
IMPLEMENTACIÓN DE UNA ARQUITECTURA
COGNITIVA PARA REDES DE SENsoRES
INALÁMBRICAS

Jose María Bermudo Mera

Julio 2014

Resumen

Una red de sensores inalámbrica es un sistema de nodos distribuidos que se encargan de interaccionar con el entorno capturando información de cualquier tipo mediante sensores, procesando dicha información, y comunicándose con otros nodos para la toma de decisiones, que podrán llevarse a cabo por medio de actuadores. Dicha comunicación entre los nodos suele realizarse de forma inalámbrica.

Uno de los problemas relacionados con las comunicaciones en este tipo de redes es la gran saturación del espectro radioeléctrico, que afecta a la fiabilidad de las comunicaciones así como a la seguridad o al consumo, que es un aspecto fundamental en este tipo de redes donde los nodos se alimentan usualmente con baterías.

Con el objetivo de realizar la comunicación entre los nodos de una forma más eficiente nacen las redes de sensores inalámbricas cognitivas (CWSN), que son aquellas en las que se utilizan estrategias de radio cognitiva en la gestión de las comunicaciones. La radio cognitiva es un paradigma de comunicación basado en que los nodos modifiquen de forma dinámica parámetros de la comunicación como la frecuencia, la modulación o la potencia con la que transmiten, y todo esto de forma coordinada.

Las redes de sensores inalámbricas cognitivas son uno de los principales objetos de investigación dentro del Laboratorio de Sistemas Integrados. Uno de los objetivos del grupo es realizar el despliegue de un banco de pruebas, de manera que se puedan probar estrategias cognitivas en el mismo, obteniendo unos resultados más realistas que los que puede ofrecer un simulador.

En este contexto, el núcleo del trabajo es la adaptación de una arquitectura cognitiva que da soporte a la gestión de las comunicaciones al middleware de comunicaciones de un nodo cognitivo (cNGD). Dicho banco de pruebas servirá tanto para probar estrategias cognitivas sobre redes de sensores reales para investigación como para desarrollar aplicaciones para innovación.

PALABRAS CLAVE: *redes de sensores inalámbricas cognitivas, banco de pruebas, bandas ISM, cNGD, arquitectura cognitiva, redes de sensores inalámbricas, radio cognitiva.*

Abstract

A WSN (*Wireless Sensor Network*) is a system formed by distributed nodes which have the purpose of getting information of any kind from the environment. This information is obtained by sensors, processed and shared with others nodes to make decisions, which will be carried out by actuators. The communication between nodes is usually performed wirelessly.

One of the problems of these networks is the increasing radio spectrum saturation that affects the reliability of the communications, their security and their energy consumption, which is a key point in these networks because nodes are usually powered by batteries.

CWSNs (*Cognitive Wireless Sensor Networks*) appear with the goal of performing the communication between nodes more efficiently. These networks are those in which strategies based on CR (*Cognitive Radio*) are used to manage communications. CR is a novel paradigm for wireless communications in which the device can dynamically change different parameters such as the frequency, the modulation or the transmission power, while maintaining the coordination between nodes.

CWSNs are one of the research lines of the LSI (*Laboratorio de Sistemas Integrados*), a research group of the DIE (*Departamento de Ingeniería Electrónica*), within the UPM (*Universidad Politécnica de Madrid*). One of the goals of the group is to deploy a testbed for CWSNs where cognitive strategies can be proved obtaining more reliable results than in a simulator.

In this context, the goal of this work is to adapt the implementation of a cognitive architecture that supports the communication management to the firmware of the cNGD, a node with cognitive capabilities. Such node will be used both to test cognitive strategies for research and to develop applications for innovating.

KEY WORDS: *cognitive wireless sensor network, testbed, ISM bands, cNGD, cognitive architecture, wireless sensor network, cognitive radio.*

Índice

Resumen	VII
Abstract	IX
1. Introducción	1
1.1. Introducción y objetivos	1
1.2. Fases del trabajo	4
1.3. Estructura de la memoria	4
2. Redes de sensores inalámbricas cognitivas	5
2.1. Redes de sensores inalámbricas	5
2.2. Redes cognitivas	7
2.3. Redes de sensores inalámbricas cognitivas	9
2.3.1. Plataformas hardware	10
3. Estudio previo	15
3.1. Hardware del cNGD	15
3.2. Firmware del cNGD	17
3.3. Arquitectura cognitiva	18
4. Adaptación de la arquitectura cognitiva	21
4.1. Módulo cognitivo	21
4.1.1. Access	22
4.1.2. Discovery	22
4.1.3. Execution	23
4.1.4. Messenger	25
4.1.5. Optimizer	25
4.1.6. Policies	26
4.1.7. Repository	26
4.1.8. VCC	27
4.2. Firmware	28
4.2.1. Nuevas funciones en la HAL	28
5. Aplicación demostradora	31
5.1. Diseño	31
5.2. Implementación	32

6. Conclusiones y Líneas futuras	37
6.1. Conclusiones	37
6.2. Líneas futuras	38
Bibliografía	41
Lista de Acrónimos	44

Lista de Figuras

1.1.	Regiones definidas por la ITU	2
1.2.	Bandas ISM en la región 1 dentro del espectro electromagnético.	2
1.3.	Propuesta de despliegue del banco de pruebas para CWSN en el laboratorio B-105 del LSI.	3
2.1.	Algunos de los protocolos más populares en las WSN.	7
2.2.	Gráfica de la utilización del espectro hasta los 6 GHz en un área urbana del BWRC	8
2.3.	Imagen de los módulos VESNA y SNE-ISMTV.	10
2.4.	WaspMote de Libelium.	11
2.5.	Imagen del Tmote Sky.	11
3.1.	Vista detallada del cNGD.	16
3.2.	Diagrama de bloques de la adaptación software de la pila de protocolos MiWi.	17
3.3.	Arquitectura del firmware del cNGD.	18
3.4.	Arquitectura del módulo de radio cognitiva.	18
5.1.	Diagrama con las acciones que se realizan tras recibir una solicitud de cambio de canal.	34
5.2.	Diagrama con las acciones que se realizan tras recibir una respuesta a una solicitud de cambio de canal previa.	34
5.3.	Diagrama con las acciones que realiza el nodo ejecutor de la estrategia.	35

Capítulo 1

Introducción

En este capítulo se lleva a cabo una introducción al trabajo en la que, además de dar una visión global del mismo, se presentan los objetivos y contexto del mismo, las etapas en las que ha sido desarrollado, y la estructura de esta memoria.

1.1. Introducción y objetivos

Una WSN (*Wireless Sensor Network*) es un sistema de nodos distribuidos que se encargan de interaccionar con el entorno capturando información de cualquier tipo mediante sensores, procesando dicha información, y comunicándose, generalmente de manera inalámbrica, con otros nodos para la toma de decisiones, que podrán llevarse a cabo por medio de actuadores. Las WSNs son una tecnología que tiene un enorme potencial, en campos muy diversos.

Junto con el gran crecimiento que han tenido las WSN en los últimos años también han proliferado otro tipo de servicios [1], que también hacen uso de redes inalámbricas. Entre estos otros servicios, los que más han crecido son los relacionados con las comunicaciones M2M (*Machine-to-machine*), y los debidos al crecimiento del uso de los teléfonos inteligentes y las tabletas.

Todos estos dispositivos se comunican utilizando las bandas ISM (*Industrial, Scientific and Medical*) del espectro, que son las bandas sin licencia, y la mayoría además en la banda de 2,4 GHz, de forma que el espectro se convierte en un recurso compartido no sólo para las WPAN (*Wireless Personal Area Network*) entre sí, sino también con otros estándares como Wi-Fi o Bluetooth que también utilizan esta banda.

Este gran número de dispositivos conectados, y concentrados en las mismas bandas debido a la asignación ineficiente del espectro, ha hecho que haya una gran saturación en el espectro radioeléctrico. Esta saturación tiene su consiguiente impacto sobre las comunicaciones provocando un aumento de la potencia de transmisión necesaria, un descenso de la QoS (*Quality of Service*), la seguridad de las mismas, etc.

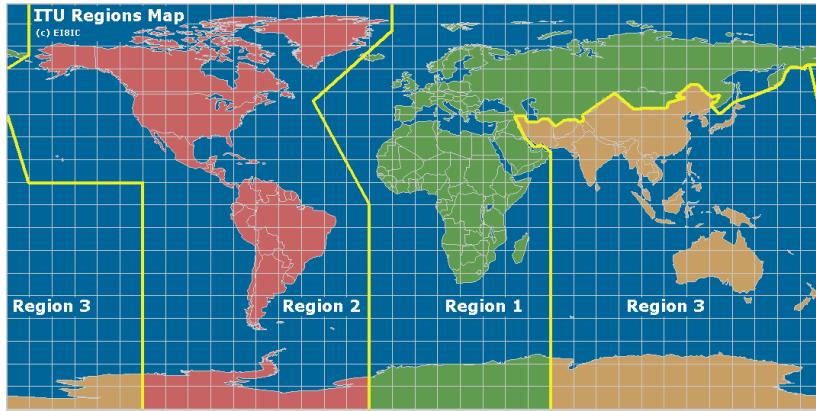


Figura 1.1: Regiones definidas por la ITU [2].

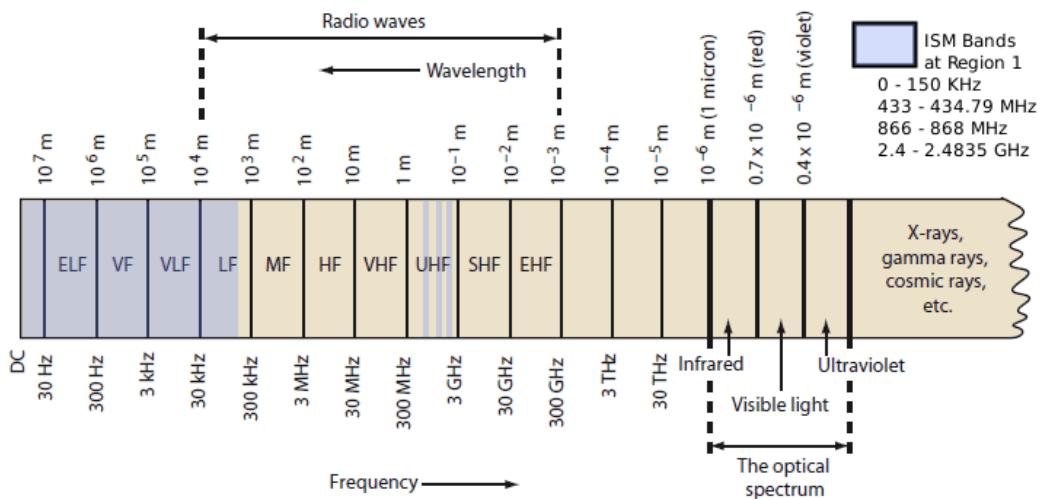


Figura 1.2: Bandas ISM en la región 1 dentro del espectro electromagnético.

Esta pérdida de fiabilidad en las comunicaciones repercute también en el consumo de los dispositivos, y este aspecto es especialmente relevante en el caso de las WSN puesto que los nodos suelen tener importantes restricciones en cuanto al consumo, ya que normalmente se alimentan con baterías.

La solución a estos problemas se ha buscado tradicionalmente mediante técnicas relacionadas con la modulación utilizada o el control de la potencia, pero este tipo de técnicas no parecen dar más de sí, por lo que han nacido otras nuevas como la CR (*Cognitive Radio*), basadas en el acceso dinámico al espectro en función de distintos parámetros del entorno. Estas técnicas han dado lugar a las CN (*Cognitive Network*), y en el caso en el que se aplican sobre las WSNs a las denominadas CWSN (*Cognitive Wireless Sensor Network*).

Las CWSN son todavía un campo en desarrollo y son uno de los principales objetos de investigación dentro del LSI (*Laboratorio de Sistemas Integrados*), grupo de investigación perteneciente al DIE (*Departamento de Ingeniería Electrónica*), en la ET-SIT (*Escuela Técnica Superior de Ingenieros en Telecomunicación*) de la UPM (*Universidad Politécnica de Madrid*).

versidad Politécnica de Madrid).

Además de ser una tecnología en desarrollo, todavía hay pocas plataformas para CWSN y pocos bancos de pruebas, por lo que hay pocos resultados en este área. Es por ello que uno de los objetivos del LSI es realizar el despliegue de un banco de pruebas [3], de manera que se puedan probar estrategias cognitivas en el mismo, obteniendo unos resultados más realistas que los que puede ofrecer un simulador.

El trabajo para el despliegue de un banco de pruebas comenzó en 2011 con el desarrollo de un primer prototipo de nodo para CWSN, el FCD (*First Cognitive Device*) [4]. El trabajo continuó con el desarrollo de un firmware que proporciona una HAL (*Hardware Abstraction Layer*) para un nodo de CWSN. Por otra parte se desarrolló un módulo para dar soporte a las estrategias cognitivas llamado CRmodule (*Cognitive Radio Module*) [5] que se basa en el modelo descrito en [6]. Y el trabajo más reciente, la implementación del cNGD (*Cognitive Next Generation Device*) [7], que es el nodo para CWSN con el que se pretende desplegar el banco de pruebas del LSI.



Figura 1.3: Propuesta de despliegue del banco de pruebas para CWSN en el laboratorio B-105 del LSI.

En este contexto, el **objetivo del trabajo es la adaptación de la arquitectura cognitiva CRmodule que da soporte a la gestión de las comunicaciones al middleware de comunicaciones existente del cNGD**.

Los subobjetivos propuestos para alcanzar este objetivo son la formación en redes de sensores inalámbricas, la familiarización con los trabajos realizados previamente en el laboratorio y con el entorno de trabajo, la adaptación de la arquitectura cognitiva, que a su vez supone la creación de nuevas funciones en la HAL y la adaptación y reestructuración de CRmodule, el desarrollo de una aplicación que sirva para demostrar la consecución del objetivo principal, y la generación de la documentación.

1.2. Fases del trabajo

El trabajo se divide en las siguientes fases:

- *Formación.*

El primer paso del trabajo es la adquisición de conocimientos relacionados con el estado del arte en redes de sensores inalámbricas y con el entorno de trabajo.

- Lectura y estudio de bibliografía y otros proyectos previos relacionados con el tema que han sido realizados en el laboratorio.
- Familiarización con el entorno de desarrollo de Microchip, MPLAB X IDE (*Integrated Development Environment*), e introducción a la programación de microcontroladores y el lenguaje C.

- *Desarrollo.*

Esta etapa consta de las siguientes tareas:

- Adaptación de la implementación de una arquitectura cognitiva para que pueda utilizarse sobre el cNGD haciendo uso de un middleware de comunicaciones y una capa de abstracción hardware implementada previamente.
- Implementación de una aplicación que funcione conjuntamente a la arquitectura cognitiva, y que servirá como demostración de la consecución de objetivos.

- *Pruebas y conclusiones.*

- Realización de pruebas sobre el nodo.
- Planteamiento de mejoras y futuras líneas de trabajo.

- *Generación de la documentación.*

La escritura de esta memoria se ha realizado mediante L^AT_EX[8].

1.3. Estructura de la memoria

Una vez realizada la introducción al trabajo vamos a explicar la estructura de este documento: En el capítulo 2 se realiza una revisión del estado del arte en la que introducimos los términos de WSN, CR y CWSN, y se habla sobre las plataformas hardware y bancos de pruebas existentes. En el capítulo 3 se caracteriza brevemente el hardware y el software sobre el que se ha realizado el trabajo. En el capítulo 4 se detalla como se ha realizado la adaptación de la arquitectura cognitiva al firmware del cNGD. En el capítulo 5 se explica la aplicación demostradora que se ha desarrollado para probar la arquitectura cognitiva sobre el nodo. Y en el capítulo 6 se recogen las conclusiones y se plantean las líneas sobre las que se puede seguir trabajando en un futuro. Para cerrar esta memoria, se incluyen unos apéndices con las referencias generadas y los acrónimos utilizados.

Capítulo 2

Redes de sensores inalámbricas cognitivas

En este capítulo se muestra un resumen del estado del arte en las redes de sensores inalámbricas así como del paradigma de la radio cognitiva y su aplicación a las mismas, dando lugar a las redes de sensores inalámbricas cognitivas. Dentro de las redes de sensores inalámbricas cognitivas vamos a hacer hincapié en las plataformas hardware y los bancos de pruebas existentes.

2.1. Redes de sensores inalámbricas

Las CWSN son sistemas de nodos distribuidos que se encargan de monitorizar variables del entorno, procesar dichos datos, comunicarse entre sí y/o tomar decisiones que podrán llevar a cabo por medio de actuadores. Los nodos que forman este tipo de redes tienen importantes restricciones en cuanto al tamaño y al coste, y además suelen estar alimentados con baterías, por lo que en muchos casos también tienen restricciones en cuanto al consumo.

El desarrollo de las redes de sensores inalámbricas comenzó en el ámbito militar, pero actualmente cubren un rango de aplicaciones muy amplio ya que es la principal tecnología sobre la que se desarrollan los ambientes inteligentes. Algunos campos en los que se aplican son:

- *Control industrial.* Las WSN se utilizan para aplicaciones como el control de procesos o la monitorización del envejecimiento de las máquinas.
- *Monitorización de grandes estructuras.* Se utilizan en infraestructuras como puentes, túneles o edificios para recoger datos que ayuden a detectar problemas derivados de las fatigas, los fenómenos meteorológicos, las vibraciones, los excesos de carga, etc.

- *Salud y telemedicina.* En este campo se plantean nuevos usos como la monitorización de parámetros corporales y la detección de patrones de comportamiento. Las WSN sirven para detectar o prevenir accidentes laborales o en el hogar, para mejorar los diagnósticos o para monitorizar el estado de los enfermos en los hospitales o en casa.
- *Monitorización medioambiental.* Los principales usos de las WSN en este campo son la monitorización de parámetros físicos como la temperatura, la humedad o calidad del aire, y la prevención y detección de desastres naturales como corrimientos de tierra o incendios.
- *Domótica.* Las aplicaciones en este campo van desde las más básicas como el control de la temperatura o la iluminación a otras más sofisticadas como la personalización del hogar en función del usuario. En numerosas ocasiones la finalidad de este tipo de aplicaciones no es sólo la comodidad del usuario, sino también el ahorro energético dentro del hogar.
- *Agricultura.* En este caso la monitorización de los parámetros físicos tiene como finalidad el control del riego o del ambiente en el caso de los invernaderos.
- *Seguridad.* Un ejemplo serían las aplicaciones de vigilancia para la detección de presencia no autorizada en una determinada zona.

Algunos de los principales aspectos a tener en cuenta en las WSN son:

- *Topología dinámica.* Las topologías que se utilicen en las redes deben ser escalables ya que el número de nodos puede variar desde decenas a cientos de nodos por cada red. Además los nodos deberán ser capaces de adaptarse por sí mismos a los cambios en la topología, manteniendo las comunicaciones operativas.
- *Funcionamiento autónomo.* Una WSN no necesita una infraestructura para funcionar ya que cada nodo cumple una función por sí mismo, pero suele existir un nodo que se encarga de reunir toda la información de la red para enviarla a un único dispositivo. A este nodo se le llama puerta de enlace.
- *Energía consumida.* Normalmente los nodos están alimentados por baterías por lo que tanto el microcontrolador como la interfaz radio y el software deben estar orientadas a minimizar el consumo, pero manteniendo un compromiso entre la autonomía del nodo y el caudal de información que puede soportar el mismo.
- *Limitaciones en el hardware.* Para lograr un bajo coste y un bajo consumo es fundamental escoger un hardware lo más simple posible, aunque esto limite la funcionalidad que puede tener el nodo, llegando a un compromiso entre funcionalidad y precio.
- *Tamaño.* Debido a la gran cantidad de nodos que se usan en las redes y a que se pretende que sean transparentes para los usuarios, el tamaño de los mismos tiende a ser cada vez más pequeño.
- *Costes de producción.* Cuanto mayor sea el número de nodos de una red más fiables serán los datos que obtengamos de la misma, por lo que será preferible que el precio unitario de cada nodo sea el más barato posible.

La mayoría de las tecnologías y protocolos de comunicación utilizados en las WSN están basados en el estándar 802.15.4 del IEEE (*Institute of Electrical and Electronics Engineers*) [9] para WPAN. El objetivo del estándar es proveer a dispositivos cercanos de una comunicación de bajo consumo sin necesidad de una infraestructura. El estándar sólo define las capas física (PHY (*Physical*)) y de acceso al medio (MAC (*Medium Access Control*)) de la torre de protocolos OSI (*Open Systems Interconnection*). Algunos protocolos basados en este estándar son ZigBeeTM [10], WirelessHART [11], ISA100.11 [12], o MiWiTM [13].



Figura 2.1: Algunos de los protocolos más populares en las WSN.

Otro estándar de comunicación inalámbrica muy popular es el 802.11 del IEEE [14], en el que se basa el Wi-Fi [15]. Es un estándar IP (*Internet Protocol*) basado en el usuario final, que no cumple con ninguna limitación de bajo consumo. Sin embargo, han aparecido nuevos estándares como 6LoWPAN (*IPv6 Low Power WPAN*) [16] que permiten encapsular paquetes IP sobre redes basadas en el estándar 802.15.4.

2.2. Redes cognitivas

Debido a la forma en la que se lleva a cabo el reparto del espectro electromagnético hay bandas como las dedicadas al uso militar, radioaficionados o comunicaciones vía satélite que están siendo infrautilizadas en gran medida, mientras que otras bandas como las bandas ISM [17] están muy saturadas. Este reparto tan ineficiente del espectro podemos verlo ilustrado en la figura 2.2. La utilización total del espectro es de un 10 % aproximadamente y sin embargo más del 95 % de este uso se concentra en las bandas por debajo de los 3 GHz.

La CR es un paradigma de comunicación radio que plantea que los dispositivos involucrados en una comunicación puedan cambiar parámetros de la misma como la modulación, la frecuencia, o la potencia, entre otros, para adaptarse mejor al entorno logrando así una comunicación más fiable y un uso del espectro más eficiente.

El concepto de CR lo introdujeron Joseph Mitola III y Gerald Q. Maguire, Jr. en 1999 [19] y posteriormente en la tesis doctoral de Mitola en 2000 [20]. Inicialmente la

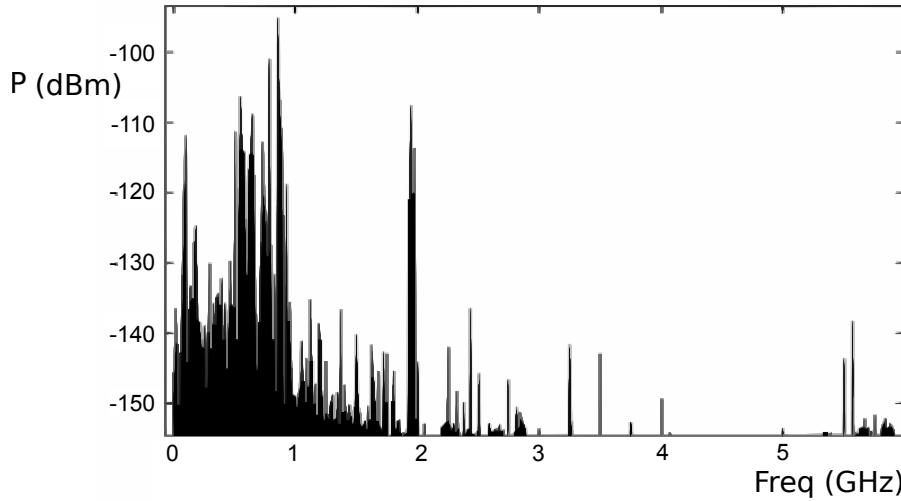


Figura 2.2: Medida de la utilización del espectro hasta los 6 GHz en un área urbana tomada a mediodía con una resolución de 20 kHz, durante 50 microsegundos, con una antena direccional de 30 grados en el Berkeley Wireless Research Center [18].

radio cognitiva se planteó como una evolución de la SDR (*Software-Defined Radio*), y una gran parte de la CR está definida por software, pero esto no tiene por qué ser necesariamente así.

La CR puede utilizarse en cualquier zona espectral, tanto en bandas bajo licencia como en bandas libres. Destacamos algunos ejemplos de estándares para CR como el que está desarrollando el grupo 802.22 del IEEE para WRAN (*Wireless Regional Area Network*), que aprovecha las bandas de televisión que no se utilizan [21], o la especificación del Grupo de trabajo 2 del 802.15 del IEEE [22].

Las principales funciones de los dispositivos de CR son: [23][24]

- *Sensado del espectro*: Es importante que sean capaces de detectar cuando hay canales libres en el espectro de forma que puedan ser utilizados para evitar interferencias con otros usuarios. Dentro del sensado del espectro se pueden distinguir además varios casos:
 - *Detección de una transmisión*: La CR debe ser capaz de determinar si hay señal en una determinada banda.
 - *Detección cooperativa*: Se refiere a técnicas en las que la información se obtiene a partir de varios usuarios [25].
 - *Detección basada en las interferencias*: A partir de la diferencia entre el nivel de ruido detectado y el máximo nivel de ruido permitido con el que un dispositivo puede establecer una comunicación. Esta técnica es la menos utilizada.
- *Compartición de la información*: Una de las características de las estrategias cognitivas es que se realizan de forma colaborativa entre distintos nodos de la red, de forma que tiene que haber algún mecanismo que dé soporte a la compartición de todos los datos sobre el estado de la red.

- *Control de la potencia:* Se utilizan técnicas que permiten ajustarse mejor a un nivel de SNR (*Signal to Noise Ratio*) a la vez que se cumplen las restricciones impuestas en cuanto a los niveles de interferencias y potencias transmitidas. En [26] podemos ver como se combinan el control de la potencia y el sensado del espectro para mejorar la capacidad de una comunicación.
- *Gestión del espectro:* La CR debe ser capaz de seleccionar la banda del espectro que mejor cumple los requisitos para comunicarse sin provocar interferencias a otros usuarios. Las funciones de gestión del espectro se realizan en dos pasos diferenciados: el análisis de espectro y la decisión de espectro.

El acceso dinámico al espectro se ha convertido en un importante campo de investigación dentro de las comunicaciones inalámbricas, y es considerado una de las claves para la red del futuro, de hecho la gestión eficiente del espectro radioeléctrico es uno de los objetivos propuestos en el H2020 (*Horizon 2020*)¹.

2.3. Redes de sensores inalámbricas cognitivas

Como hemos dicho, la mayoría de las WSN utilizan el estándar 802.15.4 del IEEE y llevan a cabo su comunicación en las bandas ISM. En la banda de 2,4 GHz, por ejemplo, este tipo de redes comparten la banda con las que utilizan el estándar 802.11, lo que da lugar a una mayor saturación del espectro, con el consiguiente aumento en las interferencias que supone una caída en las prestaciones.

Este hecho entra en contradicción con los objetivos de fiabilidad y bajo consumo característicos de las WSN, que a su vez son objetivos comunes con la CR. Esto hace que vaya creciendo el interés en utilizar técnicas cognitivas, que ayuden a solucionar estos problemas, y así es como nacen las CWSN. Las CWSN son el resultado de aplicar técnicas propias de la CR a las WSN y son la evolución natural de las mismas.

Si realizamos un repaso de la literatura podemos ver como en [6] se habla de las tendencias en este tipo de redes y de las oportunidades que ofrecen para mejorar las prestaciones globales y la capacidad de adaptación de las mismas. Además, se propone una arquitectura que se basa en agentes que implementan distintas funciones básicas (sensado del espectro, almacenamiento de la información, toma de decisiones, etc.) y mantienen un VCC (*Virtual Control Channel*), que es un canal lógico que forma parte del plano de control que utilizan los agentes para intercambiar los mensajes de gestión.

Por otra parte, en [27] podemos ver otras propuestas de arquitecturas cognitivas, mientras que en [28] se destacan algunos retos para las CWSN como la búsqueda de nuevas formas de encaminamiento que tengan en cuenta la eficiencia energética, la posibilidad de optimizar la red en función de la aplicación y la demanda de los usuarios, o la necesidad de prolongar la vida útil de los nodos.

Podemos concluir que las CWSN son todavía una tecnología en desarrollo, que necesita apoyo desde la investigación, y la implementación de estándares, simuladores y bancos de pruebas para poder obtener más resultados dentro de este campo.

¹Programa de financiación de la Unión Europea creado para apoyar y fomentar la investigación dentro del Espacio Europeo de Investigación que va desde 2014 a 2020.

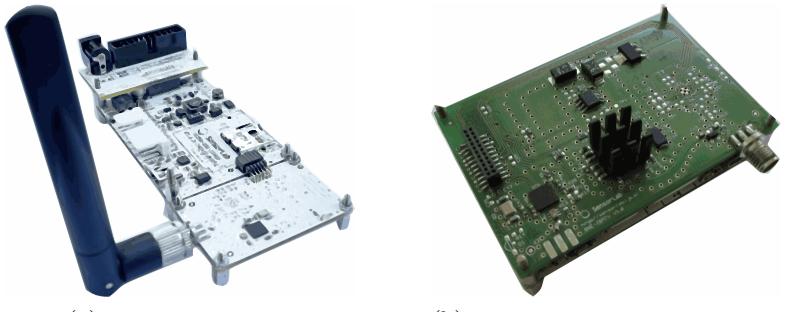
A continuación vamos a hablar de algunas implementaciones existentes, centrándonos tanto en plataformas hardware comerciales como en los bancos de prueba.

2.3.1. Plataformas hardware

El hecho de que las CWSN sean una tecnología en creciente desarrollo hace que todavía no haya muchas implementaciones hardware, y la mayoría de ellas forman parte de proyectos dentro de universidades y centros de investigación con el fin de mejorar el diseño de los propios nodos.

Entre los productos comerciales destacamos los siguientes:

- *VESNA* [29]. Es una plataforma implementada por SensorLab en el Instituto Jozef Stefan [30] para el desarrollo de WSN, está basada en ARM Cortex-M3 y dispone de una interfaz radio capaz de trabajar en distintas bandas ISM, además de otros módulos y periféricos. Soporta los estándares 802.15.4, 802.11 y 802.15.1 del IEEE, los protocolos ZigBee, 6LoWPAN, Bluetooth, Wireless M-Bus, la conexión a internet a través de Wi-Fi, Ethernet o GSM/GPRS, y el sistema operativo Contiki².



(a) Imagen de VESNA.

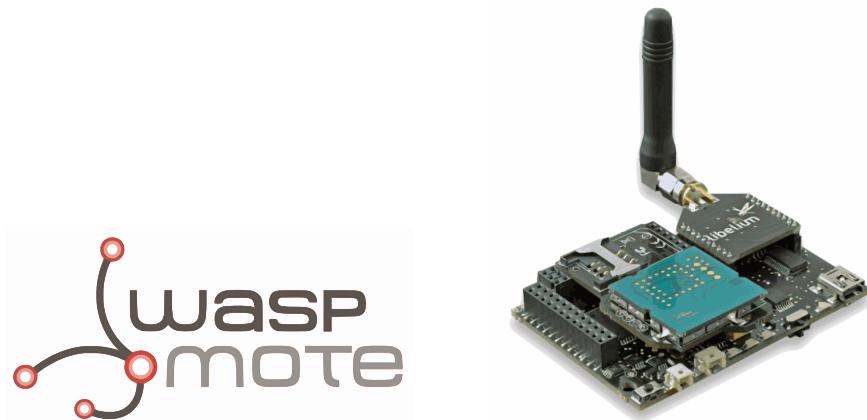
(b) Imagen de SNE-ISMTV-UHF montada sobre un nodo VESNA.

Figura 2.3: Imagen de los módulos VESNA y SNE-ISMTV.

Para dotar a VESNA de capacidades de CR se ha desarrollado SNE-ISMTV [31], una interfaz radio que dependiendo de la versión puede trabajar en parte de la banda UHF, en 868 MHz o en 2,4 GHz. Sin embargo, la interfaz radio sólo soporta cambios de canal dentro en una misma banda, por lo que no es una solución definitiva para la investigación en CWSN.

- *WaspMote*. Es un nodo basado en el microcontrolador ATmega1281 de ATMEL que ha sido desarrollado y comercializado por Libelium. Está pensado para que sirva para la mayoría de WSN que utilizan un protocolo comercial, pero sólo soporta dos interfaces radio (en caso de utilizar una placa de expansión) elegidas entre 802.15.4, ZigBee, Bluetooth, RFID, Wi-Fi, 3G/GPRS (*General Packet Radio Service*), 868 MHz y 900 MHz, por lo que es una solución limitada para la investigación en CWSN.
- *Meshlium*. También es de Libelium. Puede tener hasta cinco interfaces radio distintas, Wi-Fi a 2,4 GHz, Wi-Fi a 5 GHz, 3G/GPRS, Bluetooth y ZigBee, pero

²Contiki es un sistema operativo libre orientado especialmente a comunicaciones inalámbricas de bajo consumo y al “Internet de la Cosas”.



(a) Logo de WaspMote.

(b) Imagen del WaspMote de Libelium.

Figura 2.4: WaspMote de Libelium.

está pensado únicamente para actuar como nodo pasarela, por lo que no podemos considerarlo como una solución válida para CWSN.

- *TmoteSky*. Es un nodo de muy bajo consumo basado en el microcontrolador MSP430 de Texas Instruments. Tiene sensores de temperatura, humedad y luminosidad y una interfaz USB, además de que soporta TinyOS³. El inconveniente es que tiene un único transceptor, en la banda de 2,4 GHz, por lo que tampoco es una solución completa para la investigación en CWSN.



Figura 2.5: Imagen del Tmote Sky.

En el caso de los bancos de pruebas, ninguno de los que vamos a mencionar incluye nodos capaces de acceder a varias bandas de frecuencia, sino distintos tipos de nodos con interfaces diferentes. Este tipo de escenarios son válidos para probar estrategias cognitivas y obtener datos reales que sirvan de realimentación, pero son susceptibles de ser mejorados añadiendo nuevas funcionalidades. A continuación exponemos los principales bancos de pruebas:

³TinyOS es un sistema operativo libre orientado a plataformas de WSN.

- *TWIST (TKN Wireless Indoor Sensor network Test-bed)* [32]. Se trata de uno de los mayores bancos de pruebas a nivel mundial. Ha sido desarrollado y desplegado por el TKN (*Telecommunication Networks Group*) de la TU (*Technische Universität*) de Berlín en uno de los edificios del campus, cubriendo una superficie de más de 1500 m² a lo largo de 3 plantas y más de 40 salas. El patrón del despliegue es muy regular, formando una rejilla con una distancia de 3 m entre nodos, y consta de 102 nodos TmoteSky [33] trabajando a 2,4 GHz y otros 102 eyesIFX [34] a 868 MHz.

La capacidad de auto-configuración de la red unida al uso de hardware estandarizado y software libre hacen que la arquitectura de TWIST sea escalable, de bajo coste y fácilmente replicable. Por otra parte, se trata de un banco de pruebas pensado para experimentar con aplicaciones con una importante componente de sensado del espectro, por lo que todos los datos que se recolectan del mismo sirven para continuar explorando el campo de la CR y la optimización del espectro.

- *VT-CORNET (Virginia Tech COgnitive Radio NEtwork Test-bed)* [35]. Se trata de un despliegue de nodos de CR a lo largo de uno de los edificios del campus principal de Virginia Tech. El despliegue consta de 48 nodos estáticos con SDR basados en USRP2⁴. Además existen nodos móviles de bajo consumo para dotar de mayor flexibilidad a la red.

La finalidad de esta plataforma es poner a prueba todo tipo de estrategias, protocolos, aplicaciones y hardware específico de CR en un entorno realista, aunque también aspectos de seguridad en la red. Las interfaces radio están basadas en SDR, por lo que no son apropiadas para una CWSN debido a su alto consumo y coste.

- *LOG-a-TEC* [36]. Se trata de un despliegue de exteriores realizado en Logatec, Eslovenia, para probar estrategias de CR en las bandas ISM y de televisión. El despliegue está formado por 50 plataformas VESNA [29] y placas VSE-ISMTV [31] agrupadas en dos clústers, uno en la zona urbana y otro en la industrial. Al igual que en la red TWIST ningún nodo puede trabajar en más de una banda de frecuencia, por lo que se quedan cortos para el objetivo de investigación en CWSN.
- *w-iLab.t*. En este banco de pruebas existe un hardware llamado EE (Environment Emulator) que permite emular el comportamiento de cualquier sensor o actuador sin tener el componente real. Cada nodo del despliegue es genérico y está equipado con uno o más nodos TmoteSky, un iNode con 2 radios Wi-Fi, un EE, y una interfaz Bluetooth. Los iNodes son dispositivos Alix 3C3 [37] que funcionan con Linux, y están equipados con Ethernet, USB (*Universal Serial Bus*), puerto serie, VGA (*Video Graphics Array*), audio y dos interfaces 802.11a/b/g. Todos estos dispositivos son nodos estándar de una WSN o módulos mini PC como es el caso de los iNodes, por lo que nos limitan a la hora de explorar todas las posibilidades de una CWSN.
- *FIT/CorteXlab - Cognitive Radio Test-bed* [38]. Este banco de pruebas está en el INSA-Lyon en Francia y está todavía en desarrollo. Utilizará la arquitectura

⁴La familia USRP (*Universal Software Radio Peripheral*) proporciona una arquitectura modular con módulos intercambiables que sirven de front-end RF.

desarrollada por SensLAB [39] e integrará nodos SDR de forma que proporcionen una plataforma para la CR distribuida. Este despliegue contará con diversas arquitecturas y nodos, de forma que se puedan probar muchas posibilidades, pero el hecho de utilizar SDR hace que no se consigan los requisitos que bajo coste y pocos recursos que debe tener un nodo de una WSN.

- *Test-bed LSI.* Este banco de pruebas está siendo desarrollado en el LSI. La plataforma hardware en la que va a estar basado es el cNGD, nodo de CWSN con acceso a 3 bandas de frecuencia diferentes. En el siguiente capítulo se explican más detalles sobre dicha plataforma.

Una vez hemos realizado un repaso del estado del arte podemos concluir que las CWSN son todavía una tecnología en desarrollo, por lo que son necesarias más implementaciones de dispositivos para CWSN así como despliegues de bancos de pruebas que nos permitan continuar investigando todas las posibilidades que ofrecen y probando distintas estrategias con el objetivo de mejorar la eficiencia espectral, minimizar el consumo, o mejorar la seguridad en las redes de sensores.

Capítulo 3

Estudio previo

El objetivo de este capítulo es presentar la plataforma sobre la que se ha realizado el trabajo, y con la que se pretende realizar el despliegue del banco de pruebas para CWSN del LSI. Hablaremos tanto del hardware del nodo como del firmware del mismo y de la arquitectura cognitiva que se ha adaptado para dicho nodo.

3.1. Hardware del cNGD

El trabajo realizado en [7] tenía por objetivo el desarrollo de un nodo para CWSN con unas determinadas especificaciones en cuanto al consumo, interfaces radio, recursos y coste. Este trabajo fin de grado se ha realizado sobre dicho nodo, que recibe el nombre de cNGD, por lo que a continuación resumimos algunas de sus características hardware más importantes:

- *Microcontrolador.* El MCU (*Microcontroller*) del nodo es el PIC32MX675F256L [40], de la familia de 32 bits de Microchip. Tiene 100 pines y algunas de sus principales características son:
 - *Memoria.* 256 KB de memoria Flash y 64 KB de memoria RAM.
 - *Modos de funcionamiento.* RUN, IDLE y SLEEP pudiendo cambiar la frecuencia de reloj entre cada modo de funcionamiento, con una frecuencia máxima de 80 MHz.
 - *Timers.* Cinco timers de 16 bits.
- *Interfaces radio.* El nodo tiene tres interfaces radio, que actúan en las bandas de 434 MHz, 868 MHz y 2,4 GHz de forma que se cubren las bandas ISM de Europa. Las interfaces de 434 MHz y 868 MHz están basadas en el transceptor MRF49XA [41], y la de 2,4 GHz es un transceptor MRF24J40MA [42]. Todas las interfaces disponen de un módulo adicional que permite activar o desactivar la alimentación de las mismas.

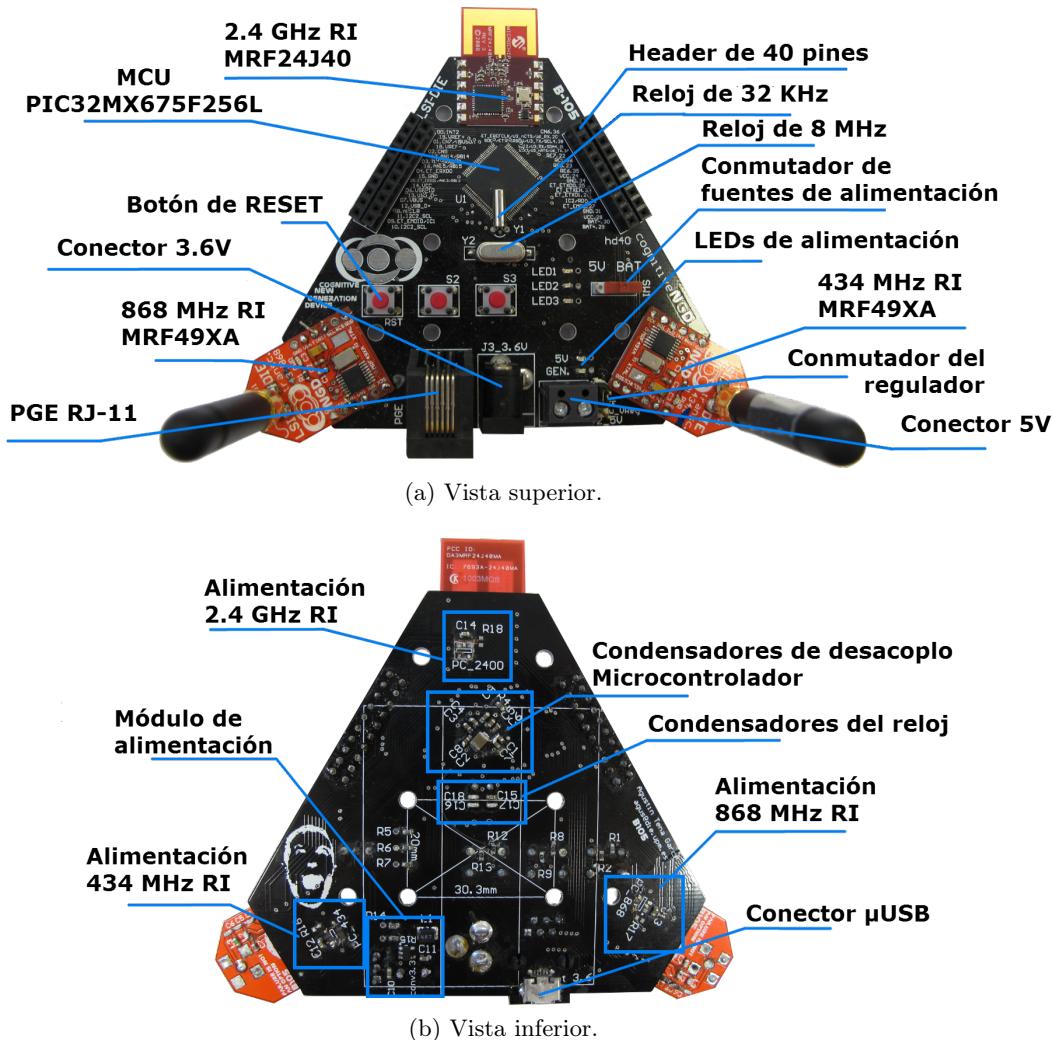


Figura 3.1: Vista detallada del cNGD.

■ *Otras:*

- *Alimentación.* Está alimentado por baterías, aunque puede alimentarse de forma cableada a través del ICD (*In-Circuit Debugger*)3, el conector de 3.6 V o el conector μ USB.
- *Módulos de expansión.* Quedan disponibles 40 pines que pueden aprovecharse para realizar módulos de expansión para el nodo. Actualmente hay dos módulos de expansión que han sido desarrollados, uno para comunicarse mediante línea serie RS232 y otro para un cargador de las baterías, y en paralelo a este trabajo se están realizando otros dos trabajos para desarrollar un módulo de *wake-on-radio* y un módulo con sensores y actuadores para el nodo.

Una vez resumidas las características del cNGD podemos ver cómo se cumplen las características hardware que debe tener un nodo para poder ser considerado un nodo para CWSN, ya que puede acceder a distintas bandas de frecuencia, en este caso a todas las bandas ISM de Europa, y además da soporte a las estrategias de bajo consumo, tanto con los distintos modos de trabajo del MCU como con la posibilidad de

cortar la alimentación de cualquiera de las interfaces radio. Además, la plataforma es flexible ya que admite cambios en los transceptores utilizados y módulos de expansión con los que implementar hardware que dé soporte a nuevas funcionalidades.

3.2. Firmware del cNGD

El firmware desarrollado en [43] tiene la función de optimizar, adaptar e integrar las pilas de protocolos de cada uno de los transceptores en una única pila, y de proporcionar una interfaz transparente para el programador con una serie de funciones que son las que acceden a los recursos físicos, en este caso a los transceptores.

Para ilustrar el resultado de la adaptación e integración de las pilas de protocolos de los tres transceptores en una única pila se muestra la figura 3.2, obtenida de [43], donde se pueden ver los distintos bloques que componen la torre de protocolos de Microchip.

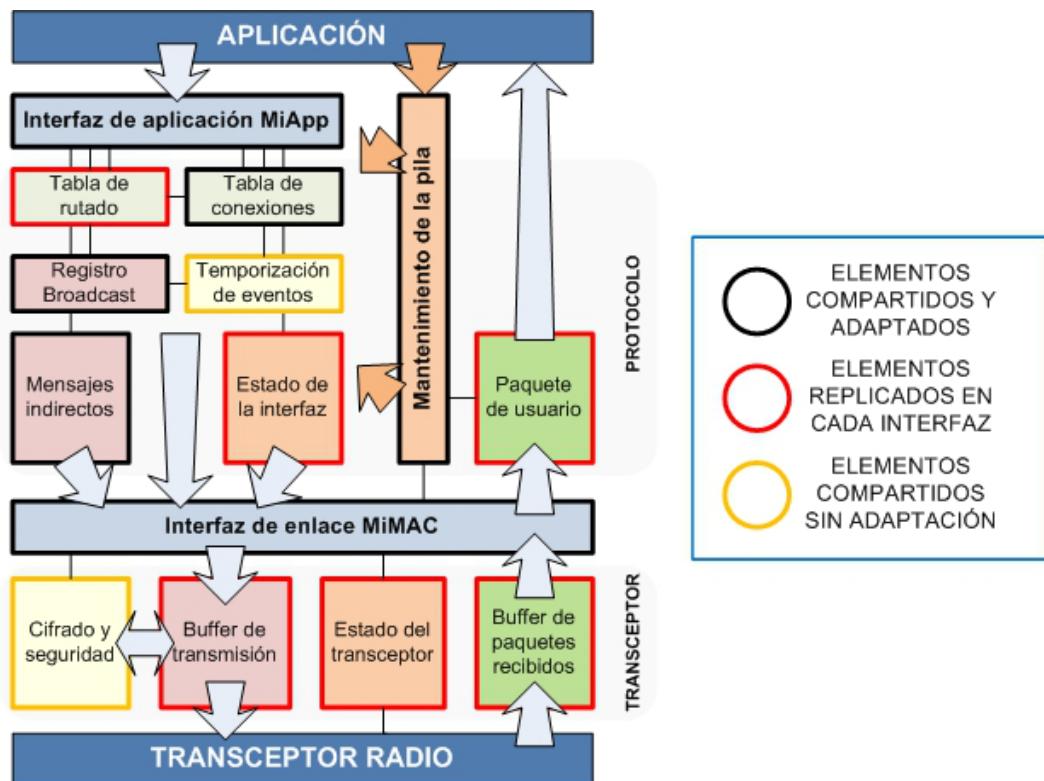


Figura 3.2: Diagrama de bloques de la adaptación software de la pila de protocolos MiWi.

Todas las funciones de acceso a recursos físicos o de la torre de protocolos forman lo que denominamos la HAL, y además de facilitar la labor del programador de la aplicación nos dan flexibilidad a la hora de cambiar cualquiera de los transceptores radio por otro, o añadir uno nuevo. Esto lo podemos ver reflejado en la Figura 3.3.

El proceso de desarrollo de la HAL se puede ver en detalle en [43], donde además de explicarse el código de errores que se utiliza en las funciones de la HAL se incluye un anexo con las funciones que ya estaban implementadas, que son las que dan soporte a las acciones fundamentales como el acceso a los flags de estado, la inicialización del nodo, la

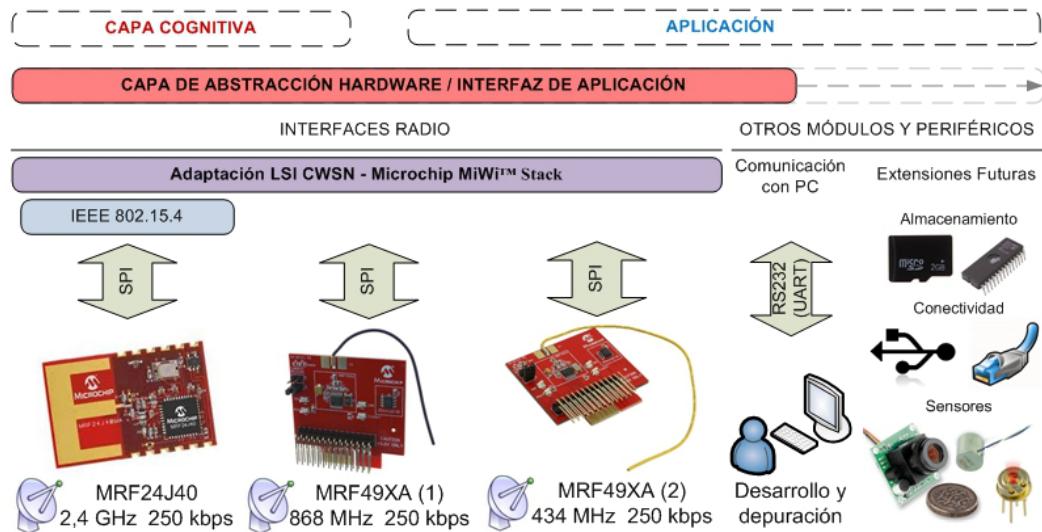


Figura 3.3: Arquitectura del firmware del cNGD.

gestión del consumo, y, por supuesto, las acciones relacionadas con las comunicaciones del nodo (envío y recepción de paquetes, acceso a las tablas de conexiones, cambio de canal, escaneo de los canales, etc.).

3.3. Arquitectura cognitiva

La arquitectura cognitiva es la parte del software que da soporte a la implementación de estrategias de CR en el nodo. El desarrollo de dicha arquitectura puede verse en [5], y supone una implementación de la arquitectura propuesta en [6] aunque realizando algunos cambios para mejorarla. Para dar un esquema general de dicha arquitectura mostramos la Figura 3.4.

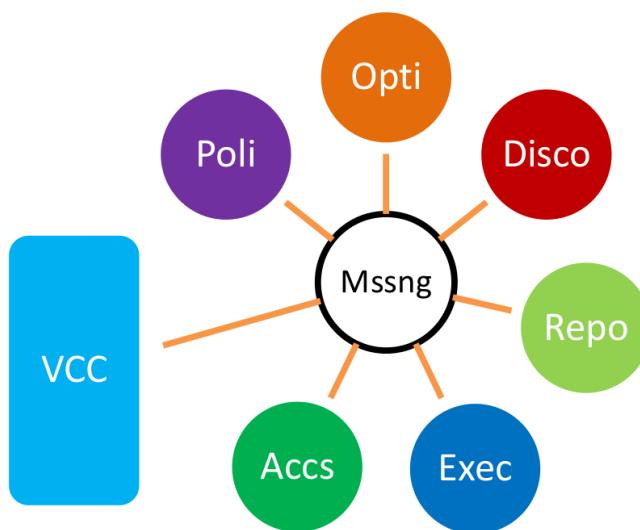


Figura 3.4: Arquitectura del CRModule, obtenida de [5].

Las funciones que debe ser capaz de realizar la arquitectura se pueden indentificar con los siguientes módulos:

- *Repository.* Un requisito para la cooperación y la colaboración eficaz es que los nodos almacenen la información aprendida, las decisiones tomadas y el estado actual para que toda esta información pueda ser consultada y compartida cuando así se requiera. Esto es posible gracias a una estructura de repositorio distribuido.
- *Discovery.* Otro aspecto fundamental es la capacidad de aprender de forma activa de las propiedades del entorno. Este módulo se encarga de extraer y filtrar la información útil a partir de una gran cantidad de datos, que pueden ser obtenidos desde diferentes niveles y capas. La información resultante se almacena en el repositorio.
- *Optimizer.* La información obtenida puede usarse para optimizar los sistemas para que se cumplan los objetivos de rendimiento u otros criterios marcados por las normas y políticas del nodo. Las optimizaciones a menudo se implementan de forma distribuida y por lo tanto cada CAgent deben ser capaz de participar en las estrategias de optimización.
- *Execution.* Los resultados del proceso de optimización necesitan ser ejecutados de forma fiable. La toma de decisiones sobre los sistemas distribuidos puede tener que realizar acciones coordinadas relativamente complejas, por lo que este módulo lleva a cabo una función fundamental. Lleva a cabo las acciones sobre los diferentes parámetros radio.
- *Access.* La forma de recopilar y difundir la información por el sistema y la participación activa en el proceso de automatización están sujetas a normas y mecanismos de confianza. Los sistemas cooperativos pueden convertirse en un objetivo para atacantes y la seguridad y el control de acceso son aspectos que deben ser abordados de forma explícita. Este módulo realiza las funciones de autenticación de los diferentes CAgents y los procesos de control de acceso a datos del repositorio.
- *Policies.* Las políticas establecen los límites y las reglas básicas de los procesos de optimización. En contraste con las prácticas actuales en las que las políticas son fijas, aquí pueden variar de forma dinámica y las redes deben ser capaces de adaptarse a estas variaciones.

El trabajo descrito en [5] no ha sido desarrollado para funcionar sobre el firmware que hemos mencionado en el apartado anterior sino que está implementado directamente sobre la torre de protocolos de Microchip y para funcionar sobre una plataforma hardware distinta, como es el FCD que es un prototipo de nodo para CWSN que fue desarrollado previamente [4].

Capítulo 4

Adaptación de la arquitectura cognitiva

En este capítulo hablaremos de los cambios realizados en cada uno de los submódulos de la arquitectura cognitiva para adaptar la misma al firmware del cNGD, explicaremos los cambios realizados en el propio firmware y se detallarán las nuevas funciones creadas en la HAL.

4.1. Módulo cognitivo

La arquitectura original del CRmodule estaba pensada para aplicar la estrategia cognitiva sobre una única interfaz radio, en concreto utilizaba la interfaz de 2,4 GHz del FCD. Sin embargo, uno de los objetivos del trabajo es que el cNGD pueda aplicar las estrategias sobre cualquiera de sus tres interfaces radio.

Por tanto, para realizar la adaptación de dicha arquitectura:

- Por un lado, hay que cambiar la implementación de las funciones de cada uno de los módulos para que utilicen funciones de la HAL en vez acceder a variables y funciones de la pila original de MiWiTM. Para realizar este cambio se utiliza tanto la refactorización de funciones como la generación de nuevas funciones en la HAL que serán usadas en dichos módulos.
- Por otro lado, será necesario realizar una reestructuración de la propia arquitectura para tener en cuenta el transceptor sobre el que se realizan las acciones de gestión.

Con este objetivo, en las siguientes subsecciones realizamos una breve descripción de la funcionalidad de cada uno de los submódulos de CRModule y de los cambios que se han llevado a cabo en cada uno de ellos. En algunas ocasiones estos cambios han requerido la creación de nuevas funciones en la HAL, por lo que hemos decidido mostrar estas nuevas funciones reunidas en 4.2.1.

Por otra parte, respecto a la implementación del software se ha mantenido la misma estructura de directorios para fuentes y cabeceras, pero se han eliminado todos los directorios y ficheros ajenos al propio CRModule ya que las variables y funciones auxiliares que se definían en ellos ahora serán funciones de la HAL, mientras que la definición de los tipos necesarios para CRModule se han reunido en la cabecera `CRModule.h`, que será la que importen ahora cada uno de los submódulos de CRModule en lugar de realizar múltiples importaciones.

Pasamos a detallar las modificaciones en cada uno de los módulos:

4.1.1. Access

Una de las características de la estrategia cognitiva es que esta se lleva a cabo de forma colaborativa entre todos los nodos de la red. Sin embargo, el hecho de que un nodo pueda realizar acciones sobre otros nodos de la red puede plantear problemas de seguridad. Así pues, la finalidad de este módulo es la de controlar qué nodos tienen permisos para realizar según qué acciones sobre el nodo receptor del mensaje de gestión.

Este módulo gestiona las tablas de permisos de cada uno de los agentes de la arquitectura cognitiva, y proporciona un método para comprobar los permisos ante una petición concreta, `BOOL CRM_AccCtrl_CheckPerm (ACCCTRL_MSSG_RCVD *Peticion)`, y otro método para añadir nuevas entradas a la tabla de permisos, `BOOL CRM_AccCtrl_AddEntry (ACCCTRL_MSSG_RCVD *Peticion)`.

El cambio realizado en la implementación de este módulo está en la comprobación de direcciones para decidir si el nodo que realiza la petición tiene permisos en caso de tratarse de una petición de chequeo de permisos, o si ya existen una entrada en la tabla de permisos en caso de tratarse de la creación de una nueva entrada en la tabla de permisos. Esta comprobación ahora se realiza con la función `BOOL isSameAddress (INPUT BYTE *Address1, INPUT BYTE *Address2)` del firmware del nodo en vez de con la función creada en la HAL de la implementación original.

4.1.2. Discovery

Este módulo es uno de los que más funciones de acción puede llegar a tener debido a su rol dentro de la arquitectura, que es el de llevar a cabo la caracterización de los distintos parámetros del entorno mediante mediciones del nivel de señal, escaneo de los niveles de ruido en cada canal, etc.

Aunque este módulo puede tener tantas funciones de caracterización del medio como parámetros del mismo se pretendan medir, las dos funciones que están implementadas actualmente son `CRM_Disc_SignalDetection`, que se encarga de realizar un escaneo del nivel de señal en cada canal, y `CRM_Disc_ActiveScan`, que realiza un escaneo activo, que consiste en enviar un paquete de saludo por cada uno de los canales, de forma que pueda servir para descubrir nuevos nodos.

Como se puede ver, ambas funciones acceden a recursos relacionados con las interfaces radio del nodo por lo que necesitamos especificar el transceptor sobre el que se realiza cada una de las acciones. Para ello vamos a necesitar el parámetro `Transceiver` que se incluye en la estructura de los mensajes que se envían y recepcionan por este módulo, por lo que dicha estructura quedaría como se muestra a continuación:

```

typedef struct _DISC_MSSG_RCVD
{
    INPUT BYTE OrgModule;
    INPUT DISCACTION Action;
    INPUT radioInterface Transceiver;
    INPUT void *Param1;
    INPUT void *Param2;
    INPUT void *Param3;
    OUTPUT void *Param4;
} DISC_MSSG_RCVD;

```

De esta forma el transceptor sobre el que se realiza la acción está disponible en las funciones de este módulo para ser utilizado cuando sea oportuno en su implementación, que detallamos a continuación para cada una de las dos funciones mencionadas anteriormente:

- ***BYTE CRM_Disc_SignalDetection (DISC_MSSG_RCVD *Peticion)***: Esta función tiene por objetivo realizar un escaneo de todos los canales para calcular el menos ruidoso y devolver dicho valor. Para implementarlo almacenamos el canal actual, que obtenemos con *BYTE GetOpChannel (radioInterface ri)*, realizamos el escaneo de los canales con *BYTE DoChannelScanning (radioInterface ri, BYTE *storeInfoHere)* conforme a los parámetros que se nos pasan en el mensaje a este módulo en *Transceiver (ri)* y **Param4 (*storeInfoHere)*, y restauramos el canal original con *BYTE SetChannel (radioInterface ri, BYTE channel)* antes de devolver el resultado del escaneo.
- ***BYTE CRM_Disc_ActiveScan (DISC_MSSG_RCVD *Peticion)***: No existía ninguna función de la HAL que permitiese llevar a cabo un escaneo activo ya que este tipo de acción en principio sólo iba a llevarse a cabo durante la inicialización del nodo para descubrir los demás nodos de la red. Por tanto se ha creado una función, *BYTE PerformActiveScan (radioInterface ri, INPUT BYTE ScanDuration, INPUT DWORD ChannelMap)*, que se encarga de realizar este tipo de escaneo conforme a los parámetros, que se le pasan en *Transceiver (ri)*, **Param2 (ScanDuration)* y **Param1 (ChannelMap)* respectivamente, y devuelve el número de respuestas válidas recibidas que se almacenan en la variable *ActiveScanResults*, y que podrá ser consultada más adelante si así se requiere.

4.1.3. Execution

Este módulo hace las veces de actuador dentro de la arquitectura puesto que es el que se encarga de llevar a cabo las acciones que se han decidido ejecutar en el módulo Optimizer como consecuencia de la ejecución de la estrategia cognitiva. Estas acciones puede ser de lo más diversas como dormir el nodo, cambiar la potencia de transmisión, cambiar el canal dentro de una interfaz, dormir o despertar alguna de las interfaces radio, etc.

En la implementación original de este módulo se distinguen las funciones que actúan sobre el microcontrolador y el nodo en general, identificadas con el sufijo MCU, de las que actúan sobre alguna de las interfaces radio, que no tienen ningún sufijo que las identifique.

Asimismo, en el diseño original ya se consideró la posibilidad de que las decisiones se pudiesen tomar sobre una interfaz en concreto por lo que ya existe un campo `Transceiver` al que sin embargo hay que cambiar el tipo para que sea de tipo `radioInterface` que es como identifican las funciones de la HAL este parámetro. Por tanto, la estructura de los mensajes será la siguiente:

```
typedef struct _EXEC_MSSG_RCVD
{
    INPUT BYTE OrgModule;
    INPUT EXECACTION Action;
    INPUT radioInterface Transceiver;
    INPUT BYTE Action2;
    INPUT BYTE Param1;
    INPUT void *Param2;
    INPUT void *Param3;
    OUTPUT void *Param4;
} EXEC_MSSG_RCVD;
```

A continuación mostramos cambios que se han realizado en las implementaciones de las funciones de este módulo, y además podemos ver que todas necesitan el campo para el transceptor que se ha creado en los mensajes para este módulo ya que las funciones de la HAL que se utilizan así lo requieren:

- ***BOOL CRM_Exec_ChanHoping (EXEC_MSSG_RCVD *Peticion):*** Esta función es la encargada de llevar a cabo el cambio de canal. Para ello se usa la función `BYTE SetChannel (radioInterface ri, BYTE channel)` de la HAL con la interfaz radio y el canal que se pasan en el mensaje destinado a este módulo en `Transceiver` y `Param1` respectivamente. Se devuelve un `BOOL` indicando si se ha conseguido realizar el cambio de canal o no.
- Además se han creado las siguientes funciones nuevas en este módulo:
 - ***BOOL CRM_Exec_SleepMCU (EXEC_MSSG_RCVD *Peticion):*** Su función es dormir al nodo, por lo que se ha aprovechado la función `BYTE SleepNode (radioInterface forceWakeUp, UINT32 slpTime_ms)` de la HAL para implementarla. En general la mayoría de las funciones de la HAL tiene implementado un código de errores de forma que en el `BYTE` devuelven `NO_ERROR` o el código de error oportuno. Dado que la función de nuestro módulo devuelve un `BOOL`, se procesa el error que nos devuelve la función de la HAL para devolver `TRUE` o `FALSE`.
 - ***BOOL CRM_Exec_TxPower (EXEC_MSSG_RCVD *Peticion):*** Se ha decidido aprovechar una función de la HAL que nos permite fijar la potencia de transmisión, `BYTE SetTXPower (radioInterface ri, BYTE powerOutput)`, para implementarla, haciendo uso de los parámetros `Transceiver` y `Param1` que se pasan en el mensaje destinado a este módulo y procesando el código de error devuelto por la HAL para devolver `TRUE` o `FALSE` si no se ha producido ningún error.

- ***BOOL CRM_Exec_Sleep (EXEC_MSSG_RCVD *Peticion)***: Realiza la acción de dormir un transceptor concreto, y se ha implementado haciendo uso de la función *BYTE SleepRadioInterface (radioInterface ri)* de la HAL. También se procesa el código de error de dicha función y en lugar de un BYTE se devuelve un BOOL indicando si se ha podido llevar a cabo la acción o no.
- ***BOOL CRM_Exec_Wake (EXEC_MSSG_RCVD *Peticion)***: Realiza la acción contraria a la anterior, despertar una interfaz radio, y para implementarla hemos aprovechado la función *BYTE WakeUpRadioInterface (radioInterface ri)* de la HAL. De nuevo se procesa el código de error para devolver TRUE si la acción se ha llevado a cabo con éxito o FALSE en caso contrario.
- ***BOOL CRM_Exec_TurnOff (EXEC_MSSG_RCVD *Peticion)***: Se encarga de apagar el transceptor que se le indica en el parámetro *Transceiver* del mensaje a este módulo. Para implementarla se utiliza la función *BYTE SwitchOffRI (radioInterface ri)* de la HAL, y se devuelve TRUE o FALSE si la acción se ha realizado correctamente o no.
- ***BOOL CRM_Exec_TurnOn (EXEC_MSSG_RCVD *Peticion)***: Es la función contraria a la anterior. La implementamos aprovechando la función *BYTE SwitchOnRI (radioInterface ri)* de la HAL y de nuevo se devuelve TRUE o FALSE indicando el éxito al realizar la acción.

4.1.4. Messenger

Este módulo es el que conecta al resto de módulos de la arquitectura cognitiva, y su función es la de reenviar los mensajes que se generan en cada módulo al módulo para el que están destinados. Este paso de mensajes se realiza independientemente del tipo o formato de los mensajes, por lo que en este caso no ha sido necesario hacer ningún cambio en la implementación.

4.1.5. Optimizer

Este módulo es el encargado de optimizar y coordinar todas las acciones que se llevan a cabo durante el proceso cognitivo. Es el que implementa la o las estrategias cognitivas que se han decidido utilizar, y para llevarla a cabo realiza peticiones a otros módulos o incluso a otros nodos.

El primer cambio realizado ha sido añadir un nuevo campo a la estructura de los mensajes de este módulo para incluir un parámetro de tipo *radioInterface* en el que se le pasa el transceptor. Este parámetro es necesario puesto que este módulo deberá mandar mensajes a otros módulos y muchos necesitan este campo. A continuación se muestra la estructura de estos mensajes:

```
typedef struct _OPTM_MSSG_RCV
{
    INPUT BYTE OrgModule;
    INPUT OPTACTION Action;
    IOPUT void *Param1;
```

```
IOPUT void *Param2;
INPUT radioInterface Transceiver;
INPUT BYTE *EUI Nodo;
} OPTM_MSSG_RCVD;
```

Junto con este cambio en la estructura de los mensajes del módulo se han realizado también otros de cambios en las implementaciones de algunas funciones de este módulo, principalmente para tener en cuenta el nuevo parámetro con la interfaz radio, pero cualquier cambio que se lleve a cabo en la implementación de las funciones de este módulo está relacionado con la estrategia que se haya decidido implementar, es decir, que no son cambios de la arquitectura propiamente dichos, sino propios de la estrategia en sí, por lo que no se explican aquí sino que se habla de todos estos cambios en el capítulo dedicado a la aplicación demostradora.

4.1.6. Policies

La toma de decisiones se lleva a cabo en función de los valores de ciertos parámetros, que se encuentran recogidos en este módulo. Estos valores constituyen las políticas de la arquitectura cognitiva, y son las que determinan el peso que tienen en la estrategia el consumo de energía, el ruido, la calidad de los enlaces, la seguridad, la latencia, el consumo de ancho de banda, la tasa binaria, y el mantenimiento de las conexiones.

En la implementación de este módulo se proporciona una función para consultar las políticas, *BOOL CRM_Poli_Read (POLI_MSSG_RCVD *Peticion)*, y otra para cambiarlas, *BOOL CRM_Poli_Write (POLI_MSSG_RCVD *Peticion)*. Dada la funcionalidad del módulo, es fácil darse cuenta de que su implementación no depende del firmware por lo que en este módulo no se ha realizado ningún cambio respecto a su implementación original.

4.1.7. Repository

La estrategia cognitiva no sólo basa la toma de decisiones en los parámetros del entorno, sino que aprende del pasado y del resto de nodos de la red, así que es este el módulo encargado de gestionar la información que pueda resultar de interés para la estrategia. Proporciona funciones para guardar o acceder a información, que puede ser del propio nodo, del entorno, o información proveniente de otros nodos de la red.

Los cambios realizados en este módulo están relacionados con el acceso a variables de la pila de protocolos, que ahora deberá hacerse a través de funciones de la HAL. Dado que la mayoría de las funciones de la HAL necesitan que se les pase como parámetro la interfaz radio, el principal cambio en este módulo ha sido la creación de un nuevo parámetro en la estructura de los mensajes que se envían o reciben de este módulo, de tipo *radioInterface*, para poder incluir el transceptor radio cuando el tipo de mensaje así lo requiera. La estructura de estos mensajes queda así:

```
typedef struct _REPO_MSSG_RCVD
{
    INPUT BYTE OrgModule;
    REPACTON Action;
    INPUT BYTE *EUI Nodo;
```

```

    INPUT REPODATATYPE DataType;
    INPUT radioInterface Transceiver;
    IOPUT void *Param1;
    IOPUT void *Param2;
    IOPUT void *Param3;
    IOPUT void *Param4;
} REPO_MSSG_RCVD;

```

Una vez implementado este cambio, la implementación de los accesos a las variables de la pila de protocolos se ha modificado para realizarlos utilizando funciones de la HAL, tal y como resumimos a continuación:

- El acceso al canal en uso cuando se está guardando información del nodo propio ahora se realiza a través de la función *BYTE GetOpChannel (radioInterface ri)* a la que hay que pasarle el transceptor como parámetro.
- El acceso a la dirección permanente cuando se está guardando información del nodo propio se realiza mediante la función *BYTE GetMyLongAddress (BYTE index)*.
- El acceso al identificador de PAN (*Personal Area Network*) cuando se está guardando información del nodo propio se realiza mediante la función *WORD_VAL GetPANID (radioInterface ri)* a la que hay que especificar la interfaz radio.
- La comparación entre dos direcciones cuando se va a almacenar información de otro nodo de la red en su entrada correspondiente se realiza con la función *BOOL isSameAddress (INPUT BYTE *Address1, INPUT BYTE *Address2)*.

4.1.8. VCC

Este módulo se encarga de proporcionar una capa de abstracción para la comunicación de los mensajes de control entre los nodos, de forma que la estrategia se pueda llevar a cabo de forma colaborativa. Debe ser capaz de distinguir cuando un mensaje recibido es de control para parsearlo y reenviárselo al módulo Messenger y de construir y enviar los mensajes que están destinados a otros nodos.

Los mensajes de control podrán ser enviados por cualquiera de las interfaces radio, por lo que el cambio relacionado con el procesamiento de los mensajes ha sido añadir también en la estructura de los mensajes destinados a VCC un campo más para indicar el transceptor. A continuación se muestra la nueva estructura de los mensajes:

```

typedef struct _VCC_MSSG_RCVD
{
    INPUT BYTE *DirNodDest;
    INPUT VCCACTION Action;
    INPUT void *BufferVCC;
    INPUT radioInterface Transceiver;
    INPUT void *Param1;
} VCC_MSSG_RCVD;

```

Otro cambio que se ha realizado relacionado con la estructura del módulo tiene que ver con los *offsets* que se definen para el parseo de mensajes, ya que ahora hay que incluir un campo más precisamente para identificar el parámetro del transceptor cuando sea oportuno.

Por último, las dos funciones que han sufrido cambios en su implementación son:

- ***BOOL CRM_VCC_Sender (VCC_MSSG_RCV *Peticion)***: Se encarga de enviar los mensajes de VCC destinados a módulos de otros nodos. Haciendo uso de los parámetros de interfaz radio, dirección destino y el buffer en el que está almacenado el mensaje que se quiere enviar al nodo remoto, y que se pasan en el mensaje destinado a VCC, se realiza el envío de la petición mediante la función *BYTE Send_Buffer(radioInterface ri, BYTE *Buffer, BYTE *Address)* de la HAL que ha sido creada para este propósito.
- ***BOOL CRM_VCC_MssgCreator (VCC_MSSG_RCV *Peticion)***: Se encarga de parsear el mensaje de control que se ha recibido para reenviárselo a Messenger. Para realizar el parseo utiliza los *offsets* definidos en la cabecera, por lo que ahora también se ha tenido en cuenta el campo de la interfaz radio.

4.2. Firmware

La inclusión del módulo cognitivo, CRModule, en el nodo también ha supuesto algunas modificaciones en el firmware del mismo. Estos cambios son:

- A nivel de configuración, se ha incluido la capacidad de decidir si vamos a utilizar la arquitectura cognitiva o no mediante una serie de directivas `#define` en la cabecera de aplicación `ConfigApp.h`.
- En caso de que vayamos a hacer uso de la arquitectura cognitiva, durante la inicialización del nodo, *InitNode()*, se llama a una rutina que se ha creado para que inicialice también el módulo cognitivo, dicha rutina es ***void InitCRModule (void)*** y está definida en `HardwareProfile.c` junto con la función de inicialización del nodo. Es importante señalar que esta inicialización debe llevarse a cabo después de haber inicializado la pila MiWiTM.

En dicha inicialización del módulo cognitivo, además de inicializar cada uno de los módulos de la arquitectura que sea necesario se configura un timer para llevar a cabo la ejecución de la estrategia cognitiva de forma periódica.

- La HAL está pensada para seguir creciendo a medida que sean necesarias determinadas funciones, y en este caso ha sido necesaria la creación de algunas funciones adicionales para realizar la adaptación de la arquitectura cognitiva. Estas funciones las reunimos en el siguiente apartado.

4.2.1. Nuevas funciones en la HAL

- ***BYTE GetMyLongAddress(BYTE index)***

- **Descripción:**
Función para leer la dirección permanente.

- **Parámetros:**
index - índice de la dirección.
- **Valor de retorno:**
 Un BYTE que contiene la dirección permanente.

- **BYTE Send_Buffer(radioInterface ri, BYTE *Buffer, BYTE *Address)**
 - **Descripción:**
 Función para enviar los datos contenidos en el buffer que se pasa como parámetro.
 - **Parámetros:**
ri - interfaz radio por la que se van a enviar los datos del buffer.
***Buffer** - puntero al buffer que contiene los datos que se quieren enviar.
***Address** - dirección destino a la que se quieren enviar los datos.
 - **Valor de retorno:**
 NO_ERROR o el código de error correspondiente.

- **BYTE PerformActiveScan(INPUT BYTE ScanDuration, INPUT DWORD ChannelMap, radioInterface ri)**
 - **Descripción:**
 Función que realiza un escaneo activo para localizar PANs operativas.
 - **Parámetros:**
ScanDuration - El máximo tiempo que puede estar realizando el escaneo de un único canal. El valor debe estar entre 5 y 14 y el tiempo real correspondiente en segundos se calcula con la fórmula de la especificación 802.15.4 del IEEE como:

$$960 \cdot (2^{ScanDuration} + 1) \cdot 10^{-6}$$

ChannelMap - Una máscara de 32 bits en la que cada bit representa cada uno de los canales, de 0 a 31, en los que se quiere realizar el escaneo.
ri - interfaz radio en la que se va a realizar el escaneo. NONE no es una opción válida en esta función.
 - **Valor de retorno:**
 El número de respuestas válidas al escaneo que han sido guardadas en la variable global ActiveScanResults.

- **WORD_VAL GetPANID(radioInterface ri)**
 - **Descripción:**
 Función para obtener el identificador de PAN de una interfaz radio en concreto.
 - **Parámetros:**
ri - interfaz radio de la que queremos obtener el identificador de PAN.

- **Valor de retorno:**

El identificador solicitado en formato correspondiente.

- **Nota:**

Esta función ha sido necesario crearla porque, aunque si se está utilizando el protocolo P2P este identificador es el mismo para todas las interfaces, en el caso de que se utilice el protocolo MiWiTM puede variar de una interfaz a otra.

- ***BYTE MiWi_Search4ShortAddress(radioInterface ri, INPUT BYTE *DireccionCorta, INPUT CONNECTION_ENTRY *Tabla)***

- **Descripción:**

Esta función busca a través de su dirección corta el índice de un nodo en la tabla de red.

- **Parámetros:**

ri - interfaz radio a la que se refiere la dirección.

***DireccionCorta** - puntero a la dirección de la cuál queremos saber el índice.

***Tabla** - puntero a la tabla en la que se va a buscar.

- **Valor de retorno:**

El índice de la entrada en la tabla de conexiones para la dirección solicitada, o 0xFF en caso de que todavía no exista dicha entrada en la tabla.

- **Nota:**

Esta función ha sido creada para ser utilizada al calcular el índice de la tabla de Repository cuando otro módulo quiere almacenar una determinada información.

Capítulo 5

Aplicación demostradora

El objetivo de este capítulo es explicar el diseño y la implementación de la aplicación que se ha desarrollado con la finalidad de probar el funcionamiento y depurar la implementación de la arquitectura cognitiva y determinados aspectos del *firmware*.

5.1. Diseño

El desarrollo de un aplicación demostradora tiene dos objetivos fundamentales: por un lado comprobar que todos los módulos de la arquitectura cognitiva funcionan correctamente de forma que dicha arquitectura es capaz de ejecutar una estrategia cognitiva, y por otro lado demostrar que la ejecución de una estrategia cognitiva es compatible con la ejecución de cualquier aplicación en el nodo.

Esta aplicación está diseñada para ejecutarse únicamente sobre dos nodos, y consiste en el envío de mensajes entre los dos nodos por un lado, y la ejecución de una estrategia cognitiva basada en la teoría de juegos desarrollada en [44] por otro. Ambas tareas deben realizarse de forma concurrente.

A continuación vamos a detallar y justificar todas las decisiones de diseño que se han llevando a cabo, principalmente con el fin de simplificar la implementación de la aplicación y buscando mantener un compromiso entre el realismo del escenario y la facilidad para apreciar los resultados de la ejecución de la estrategia cognitiva, pero siempre sin dejar de cumplir los objetivos del demostrador.

- Se ha implementado una única estrategia cognitiva, puesto que con implementar una estrategia que utilice todos los módulos, como es nuestro caso, es suficiente para probar todos los módulos correctamente.
- La estrategia elegida tiene como objetivo mejorar la comunicación inalámbrica, siendo su principal acción el cambio de canal de transmisión. Puesto que necesitamos ver de algún modo el efecto de la ejecución de la estrategia cognitiva, vamos a hacer esto realizando un seguimiento de los cambios de canal que se lleven a cabo.

- Sólo dos nodos van a participar en la estrategia cognitiva, ya que es el número de participantes de la teoría de juegos de la estrategia original. Aprovechando este hecho, se ha decidido también que el nodo ejecutor de la estrategia sea siempre el mismo, para simplificar la implementación del demostrador, y predeterminado, para tener más control sobre el demostrador de manera que también resulte más fácil la depuración tanto de la propia estrategia como de la aplicación.
- De nuevo con el objetivo de que resulte más fácil la realización de pruebas, se ha decidido que la ejecución de la estrategia se lleve a cabo periódicamente en lugar de llevarse a cabo cuando se detectan determinadas condiciones del entorno como ocurría en la estrategia original.
- Con el mismo motivo que la decisión anterior de tener más control de la ejecución y facilitar la comparación entre resultados empíricos y teóricos, hay variables de la estrategia cognitiva a las que se les va a asignar determinados valores controlados.
- Dado que la estrategia se realiza de forma colaborativa entre dos nodos al mismo tiempo que se ejecuta la aplicación, se hace necesario diferenciar entre mensajes de datos de la aplicación y de control para la estrategia.
- Además coexistir mensajes de datos y de control será necesario decidir cuáles son los prioritarios sobre los otros, o lo que es lo mismo, decidir si se da prioridad a la ejecución de la estrategia cognitiva o a la aplicación. Para ello se han considerado los siguientes aspectos:
 - El nodo puede tener una o más aplicaciones ejecutándose sobre él, y la duración de una determinada aplicación puede no estar acotada. Esto supone que si se le da la prioridad máxima a la aplicación frente a la estrategia cognitiva, podría darse el caso de que esta no se ejecutara nunca.
 - Por último, la estrategia está pensada para tener efectos positivos también sobre la aplicación por lo que darle prioridad a la ejecución de la estrategia sobre la aplicación también sería beneficioso en este sentido.

Teniendo en cuenta estos supuestos, la decisión ha sido que la prioridad de la estrategia cognitiva sea mayor que la de la aplicación.

Una vez hemos explicado las principales decisiones de diseño de nuestra aplicación demostradora también vamos a resumir brevemente algunos aspectos claves de su implementación.

5.2. Implementación

La parte de inicialización del nodo, tanto de los recursos necesarios para la aplicación como de la pila de protocolos, y el proceso de crear o unirse a la red estaba en su mayoría implementado en el firmware original del cNGD, y la implementación de las inicializaciones necesarias para la correcta ejecución de la estrategia cognitiva ya la comentamos en la sección 4.2 por lo que comenzamos directamente describiendo la implementación de la aplicación.

El bucle principal de la aplicación tiene dos pasos claramente diferenciados:

1. Se comprueba si hay algún mensaje disponible y en caso de que sea así se procesa. Se distinguen dos tipos de mensajes:

- *Mensaje de control.* Se copia el mensaje a otro buffer, específico para los mensajes de control, y se marca un flag para que la tarea periódica que implementa la estrategia cognitiva lo procese en su próxima activación. De esta forma se deja libre el buffer de recepción para que lo puedan seguir utilizando los mensajes de la aplicación.
- *Mensaje de datos.* La aplicación procesa el mensaje de datos para mostrar por la pantalla la información correspondiente.

Durante esta etapa de recepción, clasificación y procesado de los mensajes hay que tener en cuenta las siguientes limitaciones que se han impuesto para facilitar la gestión de los recursos compartidos:

- Para no retrasar la recepción de mensajes ni fragmentar la clasificación de los mismos, durante esta etapa se inhabilita la ejecución de la estrategia cognitiva hasta que esta termine.
- En caso de que se haya identificado el mensaje recibido como de control, se inhabilita el envío de mensajes de datos para la aplicación hasta que no se procese el mensaje de control. Este punto es especialmente importante porque el mensaje de control puede suponer un cambio de canal, o que el otro nodo haya realizado dicho cambio canal, y en este caso estaríamos enviando los mensajes por un canal distinto.

2. Se comprueba si se dan las condiciones necesarias para que la aplicación realice el envío de mensajes de datos, y en caso afirmativo se procede a realizar dicho envío.

Igual que en la etapa anterior, en esta también existen una serie de limitaciones, y de nuevo con el objetivo de facilitar la gestión de los recursos compartidos. Estas son:

- Igual que durante las operaciones relacionadas con la recepción de paquetes, durante las operaciones realizadas para enviar paquetes también se inhabilita la ejecución de la estrategia cognitiva. En este caso el motivo es que si se tiene que realizar el envío de un mensaje de control a otro nodo como resultado de la ejecución de la estrategia cognitiva, este podría sobreescribir los datos de la aplicación y esto supondría un problema para recuperarlos.
- La tasa de envío de datos de la aplicación es periódica simulando el envío de paquetes típico de una red de sensores.

Por último, la implementación de la estrategia cognitiva se lleva a cabo en el módulo Optimizer, y para ello utiliza dos funciones de acción fundamentales, propias de esta estrategia en concreto:

- ***BOOL CRM_Optm_Processor (OPTM_MSSG_RCVD *Peticion):*** Esta función es la que se encarga de procesar los mensajes provenientes de otros nodos. Se consideran dos tipos de mensajes para ser procesados, una petición de cambio de canal por parte de un nodo ejecutor de la estrategia, o la respuesta a una petición de cambio de canal enviada previamente.

- *Petición de cambio de canal.* Se consultan las políticas del nodo mediante un mensaje a Policies para ver si se acepta el cambio. Si no se acepta se envía una petición a VCC para que envíe la respuesta al otro nodo, en caso de que sí se acepte se realiza una petición a Executive para que cambie el canal y luego a VCC para que envíe la respuesta correspondiente al otro nodo. Tanto en las peticiones a VCC como a Executive, ahora es necesario especificar el transceptor en el correspondiente parámetro del mensaje. En la figura 5.1 se muestra un diagrama en el que se pueden ver estas acciones.

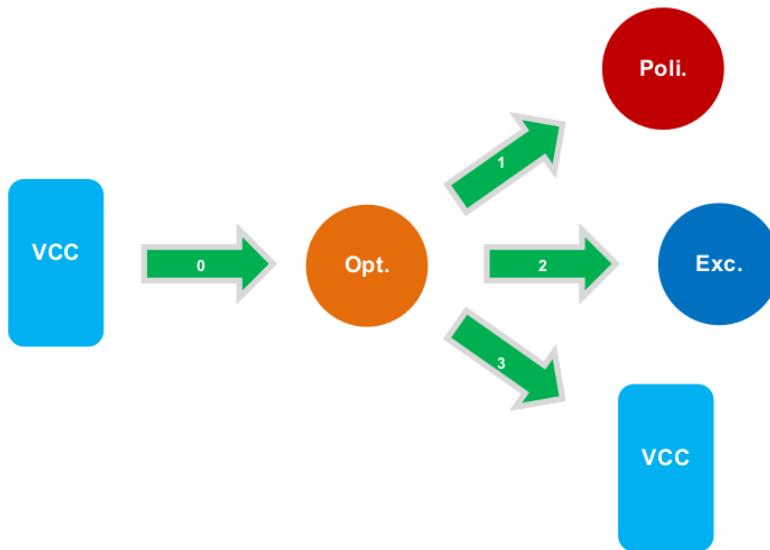


Figura 5.1: Diagrama con las acciones que se realizan tras recibir una solicitud de cambio de canal.

- *Respuesta a una petición.* Se procesa si la respuesta es afirmativa o negativa y se actualizan el número de peticiones enviadas, aceptadas y la probabilidad de cambio de canal en función de dicha respuesta. Por último se manda un mensaje al módulo Repository para almacenar esta información. En la figura 5.2 se muestra un diagrama en el que se puede ver este caso.



Figura 5.2: Diagrama con las acciones que se realizan tras recibir una respuesta a una solicitud de cambio de canal previa.

- ***BOOL CRM_Optm_GameTheory (OPTM_MSSG_RCVD *Peticion):*** Esta función es la que implementa el algoritmo en el que se basa la estrategia cognitiva que se va a llevar a cabo. Las acciones que se llevan a cabo a nivel

arquitectural se muestran en la figura 5.3, y a continuación vamos a explicar más detalladamente el procedimiento que se sigue:

El coste de cambio de canal se calcula mediante una llamada a la función ***BOOL CRM_Optm_CostAlgorithm (INPUT BYTE CosteNotChanging, INPUT BYTE CosteOcup, INPUT BYTE CosteChngChann, INPUT WORD ProbCambio Y)***, cuya implementación no ha sido modificada, y en caso de que se haya decidido cambiar de canal la secuencia que se sigue es la siguiente:

1. Se realiza una petición a Discovery para calcular el canal óptimo. En dicho mensaje hay que incluir el parámetro del transceptor en el que se quiere calcular el canal óptimo.
2. Una vez conocemos el canal óptimo se envía un mensaje a VCC para que envíe una petición de cambio de canal al otro nodo. También hay que especificar el transceptor.
3. Por último, se envía un mensaje al módulo Repository para efectuar el cambio de canal, especificando nuevamente el transceptor.

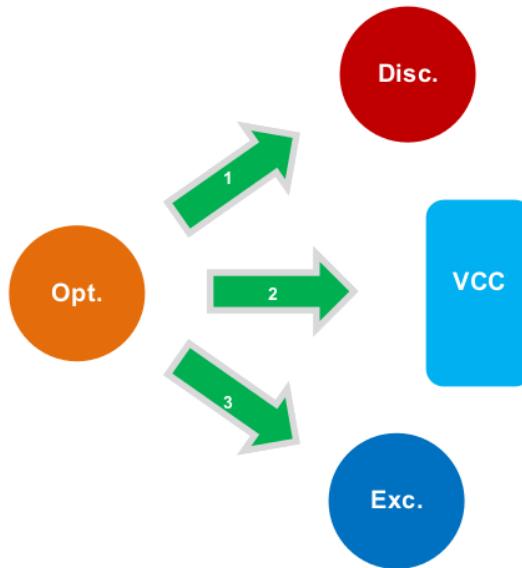


Figura 5.3: Diagrama con las acciones que realiza el nodo ejecutor de la estrategia.

Además, para implementar esta estrategia cognitiva junto a la aplicación se han realizado cambios también en la función ***BOOL CRM_Optm_Int (void)*** del módulo Optimizer, que es a la que se llama periódicamente para que realice las acciones necesarias para ejecutar la estrategia cognitiva. Estos cambios son los relacionados con la gestión de los recursos compartidos y las limitaciones que se le ha impuesto a la aplicación, que acabamos de explicar a lo largo de este capítulo.

Capítulo 6

Conclusiones y Líneas futuras

En este capítulo se presentan las conclusiones del trabajo, y se hace especial hincapié en las mejoras que podrían llevarse a cabo en el futuro.

6.1. Conclusiones

El principal objetivo del trabajo era realizar una adaptación de una arquitectura que da soporte a la implementación de estrategias cognitivas en un nodo de una WSN para que estas puedan llevarse a cabo utilizando dicha arquitectura en el cNGD. La arquitectura debe ser adaptada para ser compatible con la HAL y el middleware de comunicaciones de dicho nodo.

Para cumplir este objetivo había marcados una serie de subobjetivos, cuya consecución nos han llevado a la consecución del objetivo principal del trabajo. Estos subobjetivos están descritos en la subsección 1.2, y a continuación se muestra cómo se han ido cumpliendo:

- La formación obtenida como resultado de la lectura de la bibliografía queda patente con el repaso al estado del arte realizado en el Capítulo 2.
- Por otra parte el estudio de los trabajos previos realizados en el laboratorio queda reflejado con el breve repaso de los aspectos más importantes realizado en el Capítulo 3, y junto con la familiarización con la programación de microcontroladores y el entorno de desarrollo suponen las bases fundamentales para el desarrollo del propio trabajo.
- El subobjetivo de realizar la adaptación de la implementación de CRmodule para el cNGD, que por otro lado era la parte más importante del trabajo, también ha sido superado, y además del código generado podemos ver los resultados en el Capítulo 4.
- El otro subobjetivo relacionado con el desarrollo, que era la implementación de una aplicación demostradora, se explica en el Capítulo 5.

- Por último, la generación de la documentación en L^AT_EX que además de servir para el futuro, ha servido para aprender a utilizar esta herramienta tan potente para la edición de textos.

Una vez expuesta la consecución de objetivos, se ha decidido terminar este apartado de conclusiones señalando el hito que supone disponer de un nodo para CWSN sobre el que se puede implementar cualquier estrategia cognitiva o aplicación de una forma relativamente sencilla, gracias al soporte proporcionado por la arquitectura cognitiva y la HAL, dentro del objetivo del LSI de desplegar un banco de pruebas para CWSN.

6.2. Líneas futuras

Con vistas a continuar con el trabajo en el campo de las CWSN en el laboratorio a continuación se proponen una serie de mejoras que podrían llevarse a cabo sobre el cNGD y que abarcan tanto aspectos hardware como software:

- El primer aspecto sobre el que se puede seguir trabajando, gracias a la versatilidad de la que se dotó al cNGD en su diseño, es el desarrollo de nuevos módulos de expansión que le permitan al nodo acceder a nuevas funcionalidades. Entre estos nuevos módulos planteamos los siguientes:
 - *Nuevas interfaces de red.* Mediante una placa de expansión podríamos obtener acceso a otros protocolos de comunicación. Destacamos la posibilidad de dotar al nodo de una conexión Ethernet que sería especialmente interesante en el caso de los nodos que trabajen conectados a la red eléctrica ya que les dotaría de una mayor conectividad. Siguiendo con esta línea también podemos plantearnos el desarrollo de un módulo Wi-Fi.
 - *Memoria.* Se puede dotar de memoria a los nodos de forma que estos vuelquen todas sus trazas en dicha memoria y pasado un tiempo se puedan recoger todas las trazas para analizar los resultados desde un ordenador. La memoria podría implementarse tanto con una memoria portátil USB como con una tarjeta μSD (*micro Secure Digital*). También se podría usar esta memoria para mejorar el módulo Repository de la arquitectura cognitiva.
 - *Control remoto de los nodos.* Otra funcionalidad que se podría implementar es la posibilidad de controlar los nodos remotamente, bien desde un ordenador, o más interesante incluso desde el móvil. Para ello sería necesario una interfaz compatible, como el módulo Wi-Fi que ya se ha propuesto, un módulo de Bluetooth, o GSM (*Global System for Mobile*) en el caso de los móviles.
 - *Sensores.* Se podría desarrollar un módulo de expansión con cualquier tipo de sensores, o actuadores, de forma que sirvan para la aplicación de la propia red, o sensores más específicos destinados a mejorar la caracterización del medio radio que lleva a cabo el módulo Discovery de la arquitectura cognitiva. Con este objetivo se ha estado desarrollado un trabajo en paralelo a este en el laboratorio.
- Otro tipo de funcionalidades que se pueden implementar son las que están más relacionadas con el firmware. Proponemos las siguientes mejoras:

- *Implementación de MiWiPROTM*. Actualmente están implementados dos protocolos, P2P y MiWi, pero en caso de que el número de la red crezca mucho sería necesario implementar también MiWiPRO, que es el protocolo más complejo de los que nos ofrece Microchip.
 - *Protocolo para QoS*. Habrá ocasiones en las que sea necesario algún tipo de protocolo orientado a servicios que nos permita distinguir los tipos de paquetes para satisfacer unos determinados requisitos de QoS. Esta separación del tráfico puede ser especialmente interesante cuando coexisten distintas aplicaciones sobre un mismo nodo.
 - *Modos de seguridad*. Se podrían implementar distintos modos de seguridad, tanto a nivel de aplicación como a nivel de protocolo.
- Referidas a la arquitectura cognitiva la mejora que se propone es la siguiente:
- Se pueden mejorar y completar algunos módulos de CRmodule, como por ejemplo:
 - Replantearnos los datos que se deben almacenar y la implementación del módulo Repository.
 - Aumentar las funciones que puede llevar a cabo el módulo Discovery.
 - Implementar un mayor control en el módulo Access, de forma que no sólo se comprueben los permisos de acceso cuando los mensajes provienen de nodos distintos sino también cuando provienen de módulos distintos dentro del mismo nodo.
- Otra líneas futuras que se pueden explorar son:
- *Wake-On-Radio*. Durante el tiempo en el que una determinada interfaz radio está dormida el nodo no es consciente de la actividad que puede estar llevándose a cabo en dicha interfaz. Por otra parte, mantener despiertos los transceptores tiene un fuerte impacto en el consumo del nodo. Por este motivo, otra de las mejoras que se proponen es la implementación de transceptores con la funcionalidad de wake-on-radio, que significa que mediante un hardware adicional hay determinadas señales radio que puede despertar al nodo. En [45] podemos ver un trabajo relacionado con este tema que también ha sido desarrollado en el laboratorio.
 - *Actualizaciones OTAP (Over The Air Programming)*. Esta mejora está muy relacionada con la de *wake-on-radio*, y supone el hecho de poder programar los nodos de forma inalámbrica, lo que nos permitiría agilizar el proceso de programación y pruebas de los nodos una vez se haya realizado el despliegue, pero además, en aplicaciones reales donde puede haber nodos inaccesibles nos daría la posibilidad de reprogramar estos nodos.
Hay un trabajo que se ha desarrollado en paralelo a este en el laboratorio para llevar a cabo la implementación tanto del *wake-on-radio* como del sistema de actualizaciones OTAP.
 - *Nodo pasarela*. Con vistas a un futuro despliegue sería importante tener nodos que puedan actuar como pasarelas para la red hacia el exterior. Para este tipo de nodos podría usarse el propio cNGD con los módulos de expansión adecuados o incluso plantearse el diseño de una plataforma específica que

además de contar con las interfaces radio que se utilizan en la red contase con interfaces radio con el exterior como Wi-Fi, Ethernet u otras opciones como GPRS, GSM o 3G.

- *Estado de la batería.* Dado que los nodos están alimentados con baterías serían interesante poder consultar el estado de las mismas para conocer el tiempo de vida que le queda al nodo, o que el nodo avise a la red cuando empieza a quedarse sin batería.
- *Encapsulado del nodo.* De cara al despliegue del banco de pruebas o a una futura comercialización del cNGD, un aspecto a tener en cuenta es la necesidad de cajear el nodo.
- *Despliegue del banco de pruebas.* Por último, uno de los objetivos del LSI es realizar el despliegue de un banco de pruebas para CWSN que serviría tanto para probar estrategias cognitivas como para probar aplicaciones sobre escenarios reales.

Bibliografía

- [1] Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D., “Context Aware Computing for The Internet of Things: A Survey,” *Communications Surveys & Tutorials, IEEE*, vol. abs/1305.0982, pp. 1 – 4, May 2013.
- [2] “MapAbility for Map Resources and Overlay Mapping.” <http://www.mapability.com/>.
- [3] Romero, E.; Blesa, J.; Tena, A.; Jara, G.; Domingo, J.; Araujo, A., “Cognitive test-bed for wireless sensor networks,” *IEEE DySPAN 2014*, pp. 346 – 349, Mar 2014.
- [4] López, F., *Diseño e implementación de un nodo para redes de sensores con capacidades cognitivas, PFC*. Laboratorio de Sistemas Integrados, ETSIT-UPM, Dic 2011.
- [5] Jara, G., *Diseño e implementación de una arquitectura para la gestión de comunicaciones de una red de sensores inalámbricas cognitiva*. Laboratorio de Sistemas Integrados, ETSIT-UPM, Sep 2013.
- [6] Rabaey, J.; Wolisz, A.; Ozer Ercan, A.; Araujo, A.; Burghardt, F.; Mustafa, S.; Parsa, A.; Pollin, S.; I-Hsiang Wang; Malagon P., “Connectivity Brokerage - Enabling Seamless Cooperation in Wireless Networks,” tech. rep., Berkeley Wireless Research Center, Sep 2010.
- [7] Tena, A., *Development of a multiple RF interfaced platform for Cognitive Wireless Sensor Networks, PFC*. Laboratorio de Sistemas Integrados, ETSIT-UPM, Nov 2013.
- [8] “Latex.” <http://en.wikibooks.org/wiki/LaTeX/>.
- [9] “IEEE 802.15. WPANTM Task Group 4.” <http://grouper.ieee.org/groups/802/15/pub/TG4.html>.
- [10] “Zigbee Alliance.” www.zigbee.org.
- [11] “HART Communication Foundation, WirelessHART Technology.” <http://www.hartcomm.org>.
- [12] “ISA, Wireless Compliance Institute. ISA100 Wireless Systems for Automation.” <http://www.isa100wci.org>.
- [13] Microchip Technology Inc., *AN1066 Microchip MiWiTM Wireless Networking Protocol Stack*. 2007.

- [14] “IEEE 802.11TM Wireless Local Area Networks.” <http://www.ieee802.org/11/>.
- [15] “Wi-Fi Alliance.” <http://www.wi-fi.org/>.
- [16] “IETF. IPv6 over Low power WPAN.” <http://datatracker.ietf.org/wg/6lowpan/>.
- [17] Howitt, I.; Gutierrez, J.A., “IEEE 802.15.4 low rate - wireless personal area network coexistence issues,” *Wireless Communications and Networking, WCNC 2003*, vol. 3, pp. 1481 – 1486, Mar 2003.
- [18] Jong Kim, S., “Dynamic Spectrum Allocation with Variable Bandwidth for Cognitive Radio Systems,” *Communications and Information Technology*, pp. 106 – 109, Sep 2009.
- [19] Mitola, J. III; Maguire, G.Q. Jr, “Cognitive radio: making software radios more personal,” *Personal Communications*, vol. 6, pp. 13 – 18, Aug 1999.
- [20] Mitola, J. III, *Cognitive Radio An Integrated Agent Architecture for Software Defined Radio, PhD. Dissertation*. Royal Institute of Technology, Kista, Sweden, May 2000.
- [21] Stevenson, C.; Chouinard, G.; Zhongding Lei; Wendong Hu; Shellhammer, S.J.; Caldwell, W., “Ieee 802.22: The first cognitive radio wireless regional area network standard,” *Communications Magazine, IEEE*, vol. 47, pp. 130 – 138, Jan 2009.
- [22] “IEEE 802.15 WPANTM Task Group 2.” <http://www.ieee802.org/15/pub/TG2.html>.
- [23] Akyildiz, I.F.; Won-Yeol Lee; Vuran, M.C.; Mohanty, S., “NeXt generation/dynamic spectrum access/cognitive Radio Wireless Networks: A Survey,” *Computer Networks (Elsevier) Journal*, vol. 50, pp. 2127 – 2159, Sep 2006.
- [24] Venkatesha, R.; Hoffmeyer, J.A.; Berger, H.S., “Cognitive functionality in next generation wireless networks,” *IEEE Communications Magazine*, pp. 72 – 78, Apr 2008.
- [25] Zhiqiang Li; F. Richard Yu; Minyi Huang, “A distributed consensus-based cooperative spectrum sensing in cognitive radios,” *IEEE Trans. Vehicular Technology*, vol. 59, pp. 383 – 393, Jan 2010.
- [26] Foukalas, F.T.; Mathiopoulos, P.T.; Karetsos, G.T., “Joint optimal power allocation and sensing threshold selection for SU’s capacity maximisation in SS CRNS,” *Electronics Letters*, vol. 46, pp. 1406 – 1407, Sep 2010.
- [27] Vijay, G.; Ben Ali Bdira, E.; Ibnkahla, M., “Cognition in wireless sensor networks: A perspective,” *IEEE Sensors Journal*, vol. 11, pp. 582 – 592, Mar 2011.
- [28] Zahmati, A.S.; Hussain, S.; Fernando, X.; Grami, A., “Cognitive wireless sensor networks: Emerging topics and recent challenges,” *Science and Technology for Humanity (TIC-STH), IEEE Toronto International Conference*, pp. 593 – 596, 2009.
- [29] “SensorLab. Hardware. VESNA.” <http://sensorlab.ijs.si/hardware.html>.

- [30] “SensorLab.” <http://sensorlab.ijs.si/>.
- [31] Solč, T., “SNE-ISMTV: VESNA wireless sensor node expansion for cognitive radio experiments,” *ISWCS 2013 - The Tenth International Symposium on Wireless Communication Systems*, p. 2, Aug 2013.
- [32] Handziski, V.; Köpke, A.; Willig, A.; Wolisz, A., “Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks,” *REALMAN '06 Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pp. 63 – 70, 2006.
- [33] Moteiv Corporation, “Tmote Sky datasheet,” Jun 2006.
- [34] Eyes Project. <http://www.eyes.eu.org/>.
- [35] Newman, T.R.; He, A.; Gaeddert, J.; Hilburn, B.; Bose, T.; Reed, J.H., “Virginia tech cognitive radio network testbed and open source cognitive radio framework,” *Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on*, pp. 1 – 3, Apr 2009.
- [36] “LOG-a-TEC by SensorLab.” <http://log-a-tec.eu/>.
- [37] “PC Engines alix3d3 product file.” <http://pcengines.ch/alix3d3.htm>.
- [38] “FIT/CortexXlab: Cognitive radio testbed.” <http://www.cortexlab.fr/>.
- [39] “Senslab - Very large scale open wireless sensor network testbed.” <http://www.senslab.info/>.
- [40] Microchip Technology Inc., *PIC32MX5XX/6XX/7XX Family Data Sheet*. 2009.
- [41] Microchip Technology Inc., *MRF49XA Data Sheet, ISM Band Sub-GHz RF Transceiver*. 2009.
- [42] Microchip Technology Inc., *MRF24J40MA Data Sheet 2.4 GHz, IEEE Std. 802.15.4TM RF Transceiver Module*. 2008.
- [43] Domingo, J., *Diseño, optimización y prueba un nodo para una red de sensores inalámbrica con capacidades cognitivas*, PFC. Laboratorio de Sistemas Integrados, ETSIT-UPM, Feb 2013.
- [44] Romero, E.; Blesa, J.; Araujo, A.; Nieto-Taladriz, O., “A game theory based strategy for reducing energy consumption in cognitive wsn,” *International Journal of Distributed Sensor Networks*, p. 9, Jan 2014.
- [45] Rodríguez, R., *Diseño e implementación de un módulo de activación por radio frecuencia para redes de sensores inalámbricas*, PFC. Laboratorio de Sistemas Integrados, ETSIT-UPM, Sep 2013.

Lista de Acrónimos

6LoWPAN....	<i>IPv6 Low Power WPAN</i>
CN.....	<i>Cognitive Network</i>
CNGD	<i>Cognitive Next Generation Device</i>
CR.....	<i>Cognitive Radio</i>
CRMODULE ...	<i>Cognitive Radio Module</i>
CWSN.....	<i>Cognitive Wireless Sensor Network</i>
DIE.....	<i>Departamento de Ingeniería Electrónica</i>
ETSIT.....	<i>Escuela Técnica Superior de Ingenieros en Telecomunicación</i>
FCD	<i>First Cognitive Device</i>
GPRS.....	<i>General Packet Radio Service</i>
GSM.....	<i>Global System for Mobile</i>
H2020.....	<i>Horizon 2020</i>
HAL	<i>Hardware Abstraction Layer</i>
ICD	<i>In-Circuit Debugger</i>
IDE.....	<i>Integrated Development Environment</i>
IEEE.....	<i>Institute of Electrical and Electronics Engineers</i>
IP.....	<i>Internet Protocol</i>
ISM.....	<i>Industrial, Scientific and Medical</i>
ITU.....	<i>International Telecommunication Union</i>
LSI.....	<i>Laboratorio de Sistemas Integrados</i>
M2M.....	<i>Machine-to-machine</i>
MAC.....	<i>Medium Access Control</i>
MCU.....	<i>Microcontroller</i>
OSI	<i>Open Systems Interconnection</i>

- OTAP *Over The Air Programming*
- PAN *Personal Area Network*
- PHY *Physical*
- QoS *Quality of Service*
- SDR *Software-Defined Radio*
- SNR *Signal to Noise Ratio*
- TKN *Telecommunication Networks Group*
- TU *Technische Universität*
- UPM *Universidad Politécnica de Madrid*
- USB *Universal Serial Bus*
- μ SD *micro Secure Digital*
- USRP *Universal Software Radio Peripheral*
- VCC *Virtual Control Channel*
- VGA *Video Graphics Array*
- WPAN *Wireless Personal Area Network*
- WRAN *Wireless Regional Area Network*
- WSN *Wireless Sensor Network*