ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Proyecto Fin de Carrera

INTEGRACIÓN DE ZigBee/6LoWPAN EN UNA RED DE SENSORES INALÁMBRICA

(Integration of ZigBee/6LoWPAN into a wireless sensor network)

Para acceder al Titulo de

INGENIERO DE TELECOMUNICACIÓN

Autor: Manuel Menchaca Paz

Julio - 2012

E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

INGENIERÍA DE TELECOMUNICACIÓN

CALIFICACIÓN DEL PROYECTO FIN DE CARRERA

Realizado por: Manuel Menchaca Paz
Director del PFC: Jesús Ibáñez Díaz
Título: "Integración de ZigBee/6LoWPAN en una red de sensores inalámbrica"
Title: "Integration of ZigBee/6LoWPAN into a wireless sensor network"

Presentado a examen el día: 16-Julio-2012

para acceder al Título de

INGENIERO DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): Pérez Arriaga, Jesús
Secretario (Apellidos, Nombre): Ibáñez Díaz, Jesús
Vocal (Apellidos, Nombre): Tazón Puente, Antonio

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del PFC
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Proyecto Fin de Carrera Nº
(a asignar por Secretaría)

AGRADECIMIENTOS

Que la vida iba en serio
uno lo empieza a comprender más tarde
¡Ay, el tiempo!...Ya todo se comprende.

Gil de Biedma.

A Jesús Ibáñez por todo el esfuerzo que me ha dedicado estos meses.

A Isla porque es la persona más importante para mí.

A mi querida familia, cuya paciencia conmigo tiende al infinito.

A mis amigos de la carrera porque son muchas cosas vividas, son muchos años ya. Especialmente a Jesús González por que sin él no estaría por aquí, como él sabe. También a Luis de la Riva, gran compañero y amigo, seguro que nos vemos pronto. Por último, no podría olvidar a José María Alonso con quien es imposible empezar más proyectos sin terminar ninguno, ten paciencia, algún día lo conseguiremos.

A la gente de SAYME: Fernando, Clara, Juanma, Lorenzo y Javi. Gracias por todo, este proyecto no habría sido posible sin vuestra ayuda (y sin los cafés de las 10:30). Es imposible citar todas las cosas que he aprendido con vosotros… Y lo bien que me lo he pasado.

A mi amigo Alberto, ya sabes: "cualquier tiempo pasado fue peor".

Por último, a mis grandes compañeros de viaje: Goikoetxea, Javier, Iñigo, Mallavia y Gabriel. Sin vosotros, tampoco entendería esto. Os diré que lo bueno está por comenzar.

Gracias.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Integration of ZigBee/6LoWPAN into a wireless sensor network

Palabras clave: ZigBee, 6LoWPAN, Red de sensores, integración, IEEE 802.15.4

Capítulo 1: INTRODUCCIÓN

Durante los últimos tiempos, las redes de sensores inalámbricas han vivido una gran expansión en el ámbito de la observación y del control remoto. Aunque en su origen tuvieron carácter militar, actualmente el número de campos de aplicación es muy variado: construcción, entornos industriales, medicina, domótica, medioambiente, etc.

Las primeras especificaciones de estos sistemas comenzaron en el año 2003 con el estándar IEEE 802.15.4. Desde entonces, numerosas empresas e investigadores de todo el mundo han contribuido a su aceptación y expansión debido, sin duda, a las innumerables posibilidades que presentan estos sistemas. Sus características de bajo coste, autonomía y facilidad de implantación en cualquier entorno con una mínima intrusión, hacen que a día de hoy siga creciendo el número de nuevas aplicaciones. Las redes de sensores inalámbricas proporcionan soluciones robustas, eficientes y económicas que contribuyen a mejorar la calidad de vida de las personas.

Sin duda, nos encontramos en un momento de auge donde la necesidad de mejorar las prestaciones y de encontrar nuevos retos se hace fundamental ante la gran cantidad de opciones disponibles en el mercado.

La empresa cántabra SAYME ofrece desde hace algunos años un producto propio en el ámbito de las redes de sensores inalámbricas basado en el estándar IEEE 802.15.4. Con el fin de incrementar la competitividad y la cuota de mercado, se pretende estudiar y desarrollar un sistema compatible con el actual que implemente otras tecnologías estandarizadas: ZigBee o 6LoWPAN. Valorar la viabilidad y los beneficios que proporciona adaptar estas tecnologías a un hardware propietario constituirá el objetivo principal del presente proyecto fin de carrera. Para ello, se propondrán los cambios software/hardware necesarios para llevar a cabo la integración y se diseñará una aplicación de prueba que muestre el funcionamiento del estándar.

En el **capítulo 2**, se realizará una descripción general de las redes de sensores inalámbricas, mostrándose las características más importantes y diferentes ejemplos de aplicación. Así mismo, describiremos las prestaciones que ofrece la actual plataforma comercializada por SAYME.

A continuación, en el extenso **capítulo 3** se realizará un estudio de las diferentes tecnologías propuestas. Como punto de partida, se llevará a cabo una breve descripción del estándar IEEE 802.15.4 donde explicaremos la importancia que tiene su empleo en las redes de sensores inalámbricas. Posteriormente se analizará el funcionamiento del protocolo propietario de SAYME y las limitaciones que motivan el estudio de otras posibles tecnologías como actualización. En primer lugar, se expondrán las prestaciones y características que ofrece el reciente estándar 6LoWPAN basado en IPv6, mostrando especial interés en la dificultad que tiene implementar actualmente un producto comercial con esta tecnología. Después, se procederá a estudiar el estándar ZigBee, analizando nuevamente las ventajas e inconvenientes que puede presentar su utilización. Por último, se realizará una comparación entre los diferentes protocolos y se propondrá ZigBee como una actualización al protocolo SWAN. Para ello, se mostrarán las cuestiones y características que han motivado la decisión.

Finalizado el estudio teórico, procederemos a describir en el **capítulo 4** el hardware disponible para la implementación de ZigBee. Expondremos las diferentes modificaciones que han sido necesarias para su implementación en placas propietarias proporcionadas por SAYME.

En el **capítulo 5**, trataremos la implementación software del protocolo. Se describirá el entorno y las herramientas utilizadas para acometer la programación. Después, se analizará el stack utilizado y se plantearán las modificaciones software que han sido necesarias para su programación. Por último, se propondrán diferentes aplicaciones que demuestren algunas de las ventajas más importantes que ofrece la tecnología ZigBee frente a SWAN, planteándose cómo deberían realizarse y qué cuestiones se han de tener en cuenta.

A continuación, en el **capítulo 6** se demostrará el funcionamiento de una aplicación ZigBee, utilizando para ello capturas realizadas en un entorno de medida.

Por último, en el **capítulo 7** se expondrán las conclusiones y se hará una introducción a posibles líneas futuras.

Capítulo 2: REDES DE SENSORES INALÁMBRICAS

En este capítulo se realizará una breve descripción de las redes de sensores inalámbricas, mostrándose sus características y algunos ejemplos de aplicación. Por último, se llevará a cabo el análisis de la plataforma SENSbee como punto de partida al estudio e implementación de nuevos protocolos y estándares.

2.1 - Introducción

Las redes de sensores inalámbricas [1] (WSNs: Wireless Sensor Networks) constituyen un hito en la observación y en el control del mundo que nos rodea. En la actualidad, las tecnologías basadas en WSNs se encuentran en un momento de evolución donde no dejan de surgir nuevos avances con el fin de mejorar las prestaciones actuales.

Una red de sensores inalámbrica consiste en la distribución espacial de dispositivos autónomos que emplean sensores para monitorizar cooperativamente condiciones físicas o ambientales, como son, por ejemplo, la temperatura, presión, gases o movimiento; y que intercambian información entre sí de forma inalámbrica mediante un protocolo de comunicación prestablecido. En la Figura 2-1 se puede observar un ejemplo de red de sensores constituida por una serie de dispositivos inalámbricos que operan conjuntamente para llevar a cabo una determinada función.

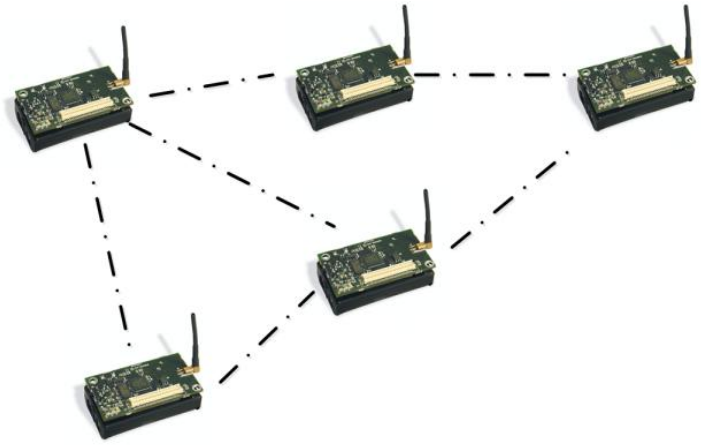

Figura 2-1: Ejemplo de red de sensores inalámbrica (WSN)

Las grandes posibilidades de estos sistemas permiten el diseño de numerosas aplicaciones de monitorización y control, que proporcionan soluciones robustas, eficientes y, sobre todo, económicas a todo tipo de problemas. Además, al ser poco intrusivos, suponen un gran avance frente a otros sistemas basados en el uso de cableado, permitiéndonos llegar a lugares que antes resultaban totalmente inaccesibles.

Su bajo coste, autonomía y fácil configuración son características que han conseguido crear un mercado y una demanda muy fuerte, donde numerosas empresas de todo el mundo trabajan para crear nuevos de productos.

Actualmente podemos encontrar redes de sensores en numerosos campos donde cobran un gran protagonismo [2], algunos ejemplos son:

- Control y monitorización industrial.
- Eficiencia energética
- Medioambiente.
- Medicina.
- Domótica
- Seguridad y seguimiento.
- Respuesta a emergencias y desastres.

Las características más importantes de las redes de sensores inalámbricas son:

- **Infraestructura de la red***:* Son redes fáciles de instalar, no requieren una configuración compleja.
- **Topología dinámica**: La topología de la red puede ser cambiante. Los nodos tienen que adaptarse ante posibles problemas para poder comunicar su información.
- **Bajos coste:** Las redes de sensores son sistemas distribuidos formados por numerosos dispositivos, de modo que resulta muy importante optimizar el máximo posible el precio unitario.
- **Consumo:** La energía constituye un aspecto muy crítico en las redes de sensores, los dispositivos que la integran se caracterizan por tener una gran autonomía.
- **Hardware:** Debe ser lo más sencillo posible, su elección repercutirá directamente en el consumo energético y en el coste económico.
- **Seguridad:** Cualquier comunicación inalámbrica debe estar protegida frente posibles ataques o intrusos.
- **Tolerancia a errores:** La pérdida de uno o varios nodos no implica que la red deje de funcionar.

Los dispositivos que forman las redes de sensores se caracterizan por tener una fuente de energía autónoma y varios módulos independientes que realizan diferentes funcionalidades. En la Figura 2-2 podemos diferenciar los módulos que constituyen estos dispositivos:

- **Transceptor:** Nos permite comunicarnos con el mundo exterior de forma inalámbrica.
- **Microcontrolador/Memoria:** Se encarga de administrar y gestionar los diferentes módulos, así como de almacenar la información necesaria.
- **Sensor/es:** Su cometido es realizar las mediciones y acondicionar las medidas tomadas.
- **Fuente de energía:** Alimentación del dispositivo.

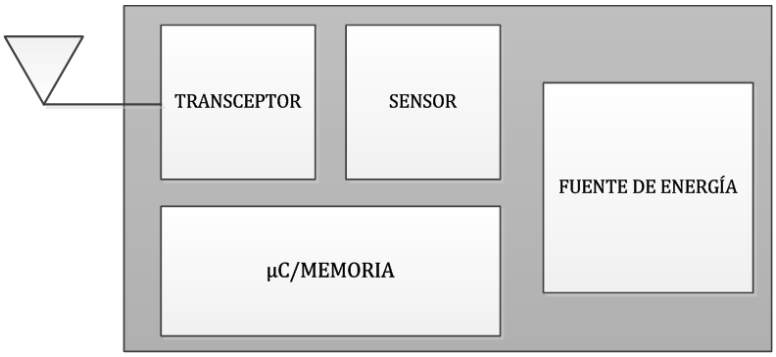

Figura 2-2: Dispositivo de una WSN

En las redes de sensores no todos los dispositivos implementan las mismas funcionalidades, por lo tanto no es necesario el mismo nivel de complejidad. El cómo gestionemos los diferentes módulos repercutirá directamente en el consumo, en las prestaciones de los dispositivos y en su coste.

2.2 - Una red de sensores: SENSbee

SENSbee [3] es una plataforma de redes de sensores inalámbricas desarrollada por la empresa SAYME Wireless Sensors para aplicaciones relacionadas con medioambiente, industria, eficiencia energética y control inteligente.

Con SENSbee podemos obtener medidas de variables distribuidas en entornos de difícil acceso, hostiles o ubicaciones remotas. También permite ejercer el control remoto de diferentes procesos de manera inteligente y autónoma.

Esta plataforma se caracteriza por su bajo consumo y excelente relación coste-prestación, ofreciendo, además, una gran modularidad y escalabilidad que permite el crecimiento de la red ante futuras necesidades. Es un producto que apuesta por la autonomía y la eficiencia energética aportando soluciones que pueden operar durante años mediante el uso de baterías. La versatilidad de la tecnología posibilita la integración del sensor que mejor resuelva las necesidades del cliente, proporcionando así soluciones específicas y muy completas.

Las características más relevantes de la plataforma son:

- **Bajo coste:** Soluciones con excelente coste/prestación para las necesidades concretas de cada cliente.
- **Modularidad y escalabilidad:** Adaptación óptima a las necesidades de cada aplicación. Facilita el crecimiento de la red ante futuras necesidades.
- **Muy bajo consumo:** Los dispositivos tienen una autonomía de más de 18 meses con dos pilas AA.
- **Integrable con sistemas existentes:** Con la posibilidad de realizar nuevos desarrollos.
- **Flexibilidad y versatilidad:** Capacidad para adaptar prácticamente cualquier tipo de sensor. Aplicación en múltiples escenarios.
- **No requiere cableado:** Permite una fácil instalación con la mínima intrusión posible.

La plataforma realiza sus comunicaciones mediante el protocolo SWAN, el cual está basado en el estándar IEEE 802.15.4. El sistema actual de la compañía SAYME cuenta con sensores acondicionados para realizar diversas medidas ambientales (temperatura, velocidad y dirección del viento, etc.) y también físicas (aceleración, peso, distancia, etc.).

Como se puede observar en la Figura 2-3, existen dos tipos de dispositivos en esta plataforma; los SPOTs y los MASTERs. En líneas generales, los SPOTs se encargan de adquirir las señales de uno o varios sensores, de su procesamiento y de la transmisión de los datos al MASTER. El SPOT dispone de un conector de expansión que permite incorporar hasta seis sensores.

Por su parte, los MASTERs centralizan toda la información procedente de los SPOTs y la transmiten a través de sus interfaces (USB, WiFi, RS232, Ethernet, GPRS, Bluetooth) al mundo exterior (Internet, por ejemplo). Así pues, la función del MASTER es la de atender a los distintos SPOTs, garantizar la correcta sincronización y proporcionar una comunicación fiable.

Figura 2-3: Plataforma SENSbee

SAYME proporciona también las aplicaciones software SENScada y SENSview para gestionar la red de forma remota y para explotar y visualizar de forma gráfica los datos capturados por los SPOTs desde un entorno Web. De esta forma, el cliente puede controlar y monitorizar su red mediante una interfaz gráfica desde su ordenador, de manera sencilla e intuitiva, a través de Internet.

Capítulo 3: PROTOCOLOS SOMETIDOS A ESTUDIO

Se describirán los protocolos IEEE 802.15.4, 6LoWPAN y ZigBee empleados en el ámbito de las redes de sensores inalámbricas. Al final del capítulo se mostrarán las conclusiones obtenidas y la decisión de qué tecnología es más adecuada para implementar en un determinado hardware propietario.

No se pretende realizar una descripción exhaustiva de los distintos estándares y protocolos, sino que se realizará una presentación somera de los mismos prestando especial dedicación a aquellos aspectos que inciden de forma directa en las prestaciones y características de una red de sensores de ultra bajo consumo y coste, como es SENSbee.

3.1 - IEEE 802.15.4

En el año 2003 el IEEE (Institute of Electrical and Electronics Engineers) desarrolló el estándar 802.15.4 [4] dentro del grupo de trabajo 4 del 802.15. Su propósito era dar servicio a un nuevo tipo de red inalámbrica de área personal, concretamente a las LR-WPANs (Low Rate Wireless Personal Area Network).

Las LR-WPANs son redes empleadas para transmitir información a distancias relativamente cortas. Están constituidas por dispositivos de bajo coste preparados para trabajar con baja tasa binaria, donde los requisitos de calidad de servicio son simples o inexistentes. Además, se caracterizan por el bajo consumo energético y porque operan en bandas de frecuencia internacionales sin licencia. Estas características son las que diferencian al IEEE 802.15.4 de otras tecnologías disponibles para las WPANs (Wireless Personal Area) como son, por ejemplo, Bluetooth o WiFi.

El IEEE 802.15.4 es un estándar que ha seguido evolucionando durante el paso de los años. Así, desde el año 2006 se realizan revisiones del IEEE 802.15.4 con el objetivo de resolver problemas relacionados con las bandas de frecuencia de trabajo y mejoras en conceptos relacionados con la seguridad de las comunicaciones. Durante los últimos años, se han estado realizando revisiones en la modulación utilizada en la banda 868/915 MHz, con el fin de mejorar las prestaciones del estándar. Con la última revisión del año 2011, se extiende el mercado de operatividad a las áreas metropolitanas y se resuelven también numerosas ambigüedades pendientes.

En la Figura 3-1 se observa que el estándar IEEE 802.15.4 sólo define el nivel físico y el control de acceso al medio de redes LR-WPAN.

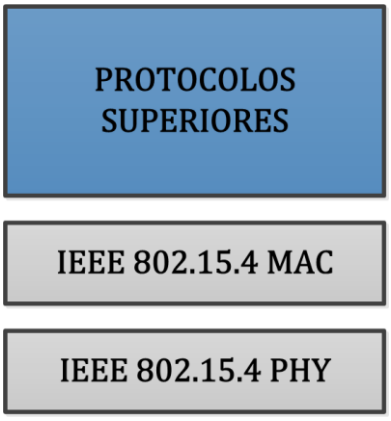

Figura 3-1: Stack 802.15.4

Las características de IEEE 802.15.4 hacen que sea un pilar básico para todos los estándares y protocolos que implementan las capas superiores de las redes LR-WPANs. Aportando un nivel físico y de enlace basado en conceptos de bajo consumo, coste y sencillez.

3.1.1 - Capa Física

La capa física proporciona un servicio de transmisión de datos y una interfaz de gestión de la propia capa. Permite dos modos de funcionamiento en función de la bandas de frecuencia en la que operemos. Así pues tenemos:

- Banda ISM (Industry Science Medicine) a 2.4 GHz.
- Banda 868 MHz en Europa ó 915 MHz en América.

En la Figura 3-2 podemos observar cómo se distribuyen los canales [5] en el IEEE 802.15.4. En la banda 915 MHz contamos con 10 canales con una separación de 2 MHz y un sólo canal para 868 MHz. Por razones de coste, se fabrican circuitos que puedan operar en ambas frecuencias debido a su proximidad. Por otro parte, en la banda de 2.4 GHz disponemos de 16 canales con una amplia separación de 5 MHz, lo cual facilita el diseño de los filtros de transmisión y recepción reduciendo considerablemente los costes.

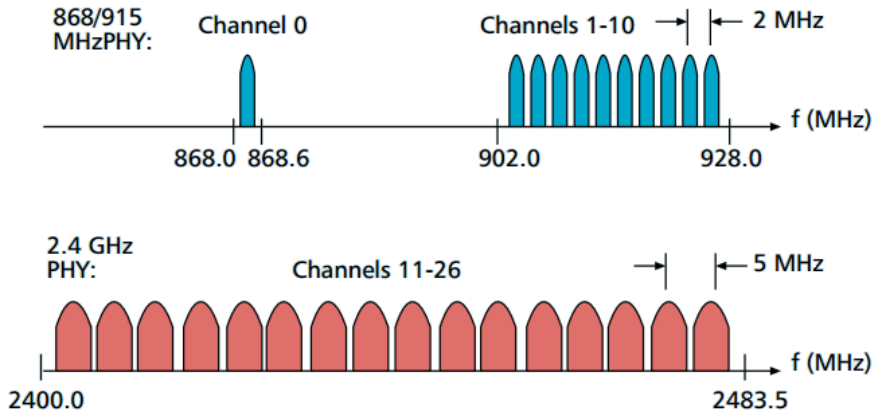

Figura 3-2: Estructura de los canales IEEE 802.15.4

Es evidente que este tipo de dispositivos no tienen una idea de movilidad entre países como podría suceder con otros sistemas como la telefonía móvil, por tanto, resulta necesario seleccionar con antelación la banda de frecuencia en la que vayamos a trabajar. Por su parte, son claras las ventajas de trabajar en la banda ISM por motivos de mercado (es una banda internacional sin licencia), pero no podemos olvidar su saturación actual debida a la convivencia con otras tecnologías inalámbricas como el Bluetooth o el WiFi.

PHY (MHz)	Banda frecuencial (MHz)	Canales	Modulación	Tasa chip (Kchip/s)	Tasa binario (Kbps)
868	868-868.56	0	BPSK	300	20
915	902-928	1-10	BPSK	600	40
2450	2400-2483.5	11-26	O-QPSK	2000	250

Tabla 3-1: Comparativa nivel físico

En la Tabla 3-1 podemos observar las características asociadas a la elección de la frecuencia de trabajo. En la banda de 2.4 GHz podemos alcanzar tasas binarias de hasta 250 Kbps, mientras que en 868/915 MHz sólo podemos trabajar respectivamente con velocidades de 20 Kbps y 40 Kbps. Esta diferencia es debida a que el esquema de modulación 0-QPSK (Offset Quadrature Phase-Shift Keying) utilizado a 2.4 GHz es de mayor orden que el BPSK (Binary Phase-Shift Keying) a 868/915 MHz, además de disponer de un mayor ancho de banda.

Aunque en 2.4 GHz es posible obtener una mayor tasa binaria, el emplear las otras bandas de frecuencia puede tener otras ventajas relacionadas con una mejor sensibilidad y una mayor área de cobertura. Este tipo de ventajas nos permitiría, por ejemplo, reducir el número de nodos necesario en un determinado área y por lo tanto el coste.

Los esquemas de modulación emplean la técnica espectro ensanchado por secuencia directa (DSSS: Direct Sequence Spread Spectrum). Mediante esta técnica se consigue aumentar el ancho de banda de la transmisión y reducir la densidad espectral de potencia. El uso del espectro ensanchado proporciona una mayor robustez frente al fenómeno de multipath[1] (multi-camino) y nos permite transmitir con una mayor potencia, siempre y cuando la limitación venga impuesta por la densidad espectral de potencia.

Para la banda de 868/915 MHz se emplea la modulación BPSK con filtros rectangulares con el fin de obtener una envolvente constante, lo cual permite el uso de amplificadores de potencia muy eficientes y de bajo coste. Se emplea además una aproximación de DSSS en la que cada bit transmitido es representado por una cadena de ensanchado de una longitud máxima de 15 bits. Los datos binarios son codificados multiplicando estas secuencias por +1 ó -1, modulando la secuencia resultante sobre la portadora empleando la modulación BPSK.

En el esquema de modulación a la frecuencia de 2.4 GHz los datos binarios son agrupados en símbolos de 4 bits cada uno, donde cada símbolo especifica una de las 16 secuencias ortogonales pseudoaleatorias (PN) de 32 chips para el proceso de transmisión. Las secuencias PN para símbolos de datos consecutivos son concatenadas, siendo la nueva secuencia total modulada sobre la portadora empleando O-QPSK con filtros cosenoidales. Este esquema, equivalente a la modulación MSK (Minimum Shift Keyng), proporciona una señal de salida con una envolvente de amplitud constante, lo que supone una enorme ventaja desde el punto de vista de la amplificación. Por tanto, nos permite emplear amplificadores de potencia muy eficientes, reduciendo costes. Además, la señal resultante posee un espectro muy confinado por lo que sus lóbulos secundarios no generan interferencias en los canales adyacentes.

El estándar define una sensibilidad de los receptores mínima, siendo de -85dBm para 2.4 GHz y de -92dBm para las bandas de 868/915 MHz. Estos valores incluyen un margen suficiente para cubrir las tolerancias de fabricación así como para permitir implementaciones de muy bajo coste, por lo que no es extraño encontrar dispositivos en el mercado que nos ofrezcan una sensibilidad del orden de 10dB mejor que la marcada por la especificación.

Los dispositivos han de ser capaces de transmitir al menos 1mW de potencia, aunque en función de la necesidad esta potencia puede ser mayor o menor, dentro de los limites regulados. Así pues el alcance mínimo es de unos 20m, pero es posible llegar hasta casi los 100m con una configuración adecuada en el receptor y en el transmisor. Aunque, como más adelante se explicará, en función de la topología que empleemos podremos aumentar el rango de cobertura sin aumentar el nivel de potencia de los dispositivos.

Los requisitos del estándar en cuanto a la calidad del servicio (QoS) son muy bajos, y se considera la opción de retransmisión cuando la comunicación falla. La comunicación es half-duplex, lo cual hace que no podamos transmitir y recibir al mismo tiempo, simplificando notablemente el diseño.

[1] Propagación multi-camino: Es el fenómeno que tiene lugar cuando las señales de radio llegan a las antenas receptoras por dos o más caminos y en diferentes tiempos. Puede causar problemas en la recepción de la señal debido a la interacción entre las señales recibidas.

3.1.2 - Capa MAC

El control de acceso al medio (MAC) proporciona dos servicios, por un lado tenemos las transmisiones de datos y por otro la gestión propia de la capa. Básicamente su tarea consiste en llevar a cabo los procesos de asociación/disociación, reconocimiento automático de tramas, mecanismos de acceso al canal, validación de trama y la gestión de beacons[2] en la red.

La trama de nivel MAC tiene un payload máximo de 127 bytes, ofreciendo la suficiente flexibilidad para cubrir las necesidades de otros protocolos que empleen el nivel de enlace del IEEE 802.15.4. Longitudes típicas para aplicaciones domésticas como el control de la seguridad o la iluminación se estima en unos 40-60 bytes, mientras que en aplicaciones de carácter industrial o multisalto puede ser mayor debido a la necesidad de direccionamientos más extensos.

Existen cuatro tipos de trama a nivel de enlace: beacon, datos, confirmación (ACK) y comandos. En función de qué tarea desempeñe la trama, podrá variar su longitud y el número campos dentro de los 127 bytes disponibles. El formato general de la trama de nivel enlace se puede apreciar a continuación en la Figura 3-3.

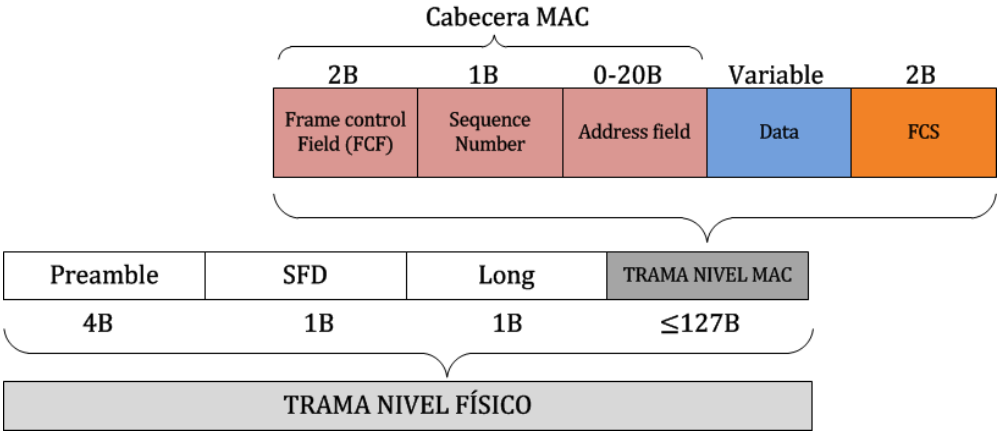

Figura 3-3: Formato trama 802.15.4

Como puede observarse, la trama MAC está compuesta por tres bloques:

- **Cabecera (MHR: MAC Header):** Consta de un número variable de bytes y en ella se encuentran los campos de control del protocolo a nivel enlace.
  - Frame Control Field (FCF): Indica la configuración (tipo de trama, formato del campo de direcciones, codificación, etc.) de la trama MAC. Además, informa si es necesario confirmar la recepción. En caso de que así sea, el receptor tras comprobar la integridad de la trama mandará automáticamente la confirmación correspondiente. Con las tramas beacon y de confirmación no es necesario.
  - Sequence number: Número de secuencia.
  - Address field: Puede incluir las direcciones de dispositivo origen y/o destino, los identificadores de las PAN de origen y/o destino, en función de la configuración establecida en el campo FCF. Su longitud varía entre 0 y 20 bytes, puesto que en determinadas ocasiones es posible omitir alguna de estas direcciones. Permite dos tipos de direccionamiento: 16 bits (short) ó 64 bits (EUI-64).
- **Datos:** Es el campo que contiene la información que estamos transmitiendo. Únicamente emplean este campo las tramas beacon y de datos, puesto que contienen información relevante de niveles superiores.

[2] Un beacon es una trama de nivel enlace que se emplea en la gestión de la red.

- **Frame Check Sequence (FCS):** Se encarga de comprobar si la trama es correcta mediante un algoritmo de redundancia cíclica de la UIT-T[3].

Además, la capa física añade a la trama MAC una cabecera con los siguientes campos:

- **Preamble:** Campo 32 bits diseñado para permitir la adquisición de símbolo y la sincronización temporal.
- **SFD (Start-of-packet delimiter):** Indica el comienzo de la trama.
- **Long:** Especifica la longitud de la trama MAC.

La capa MAC será la encargada de filtrar las tramas recibidas, rechazando todas aquellas que no sean de interés. En primer lugar se descartan todas las tramas que no verifican su FCS. Este campo se calcula en el dispositivo de origen a partir de las cabeceras y el campo de datos de la trama MAC. Al llegar ésta a su destino se vuelve a efectuar el cálculo, comparándola con el valor del FCS recibido. Si no coinciden, la trama se considera incorrecta y es descartada.

3.1.3 - Características

3.1.3.1 - Tipos de dispositivos

El estándar define dos tipos de dispositivos teniendo en cuanta su funcionalidad. Por tanto, disponemos de dispositivos de funcionalidad completa (FFD: Full Function Devices) y dispositivos de funcionalidad reducida (RFDs: Reduced Function Devices).

Los **FFDs** están equipados con todas las funciones definidas por el estándar, de modo que implementan las siguientes características:

- Pueden operar como coordinadores de la red de área personal (PAN: Personal Area Network), como simples coordinadores o como dispositivos finales (end-devices).
- Pueden comunicarse con cualquier dispositivo compatible con el estándar IEEE 802.15.4, ya sean FFDs o RFDs.
- Requieren una mayor cantidad de recursos hardware.

Los **RFDs** son necesarios debido a que no todos los dispositivos requieren el mismo nivel de complejidad. Suelen ser normalmente los sensores o actuadores de una red. Sus características más importantes son:

- Sólo pueden actuar como end-devices, nunca como coordinadores.
- Sólo se comunican con FFDs.
- Sólo se puede asociar a un único FFD.
- Su implementación resulta simple y barata, requieren menos recursos hardware.
- Suelen estar alimentados por baterías, caracterizándose por implementar técnicas para el bajo consumo.

Disponemos de las siguientes topologías de red [6]:

- Estrella.
- Peer-to-peer.
- Cluster-tree.

En la topología en estrella se define un FFD como coordinador de la PAN o nodo central. Toda comunicación tiene que pasar por el coordinador, ningún nodo (FFD o RFD) puede interactuar

[3] UIT-T es órgano de normalización de las Unión Internacional de Telecomunicaciones (UIT).

directamente con otro que no sea el nodo central. De modo que todas las comunicaciones están centralizadas.

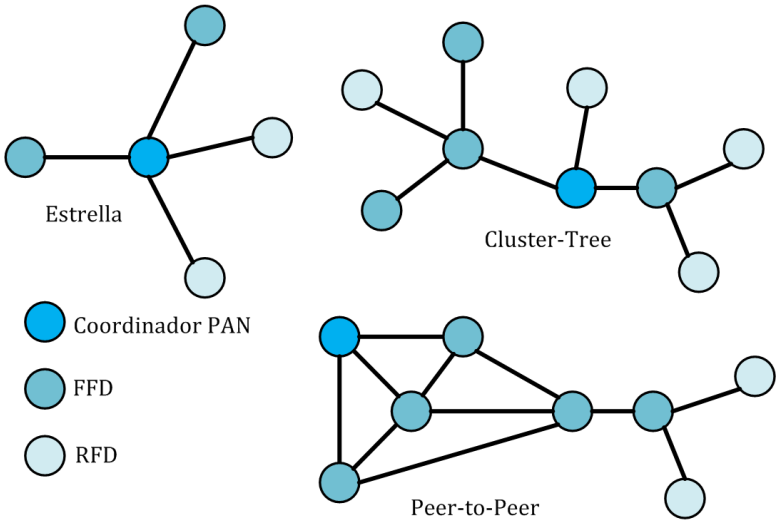

Figura 3-4: Topologías 802.15.4

En la topología peer-to-peer también existe un coordinador de la PAN, pero en este caso están permitidas las comunicaciones entre cualquier dispositivo de la misma red. Por tanto, ahora podrán existir comunicaciones intermedias que ayuden a alcanzar a un determinado nodo. Sólo los FFDs tienen la capacidad de poder decidir qué camino es el más óptimo, permitiéndose así las comunicaciones con múltiples saltos entre un origen y un destino. Por último, tenemos la topología cluster-tree que como podemos observar en Figura 3-4, dispone de un coordinador de red global (raíz) al que se conectan otros nodos. Algunos de estos nodos, puede actuar como coordinadores locales (FFDs), gestionando a su vez otros nodos simples (FFDs o RFDs). No obstante, siempre deberá existir un coordinador global que gestione el funcionamiento de la red.

3.1.3.2 - Modos de funcionamiento de la red

El nivel de enlace define los siguientes modos de funcionamiento de la red:

- **Modo Beacon (beacon-enabled network):** El coordinador de la red envía periódicamente tramas beacon para sincronizar los nodos que forman parte de su red. El intervalo temporal comprendido entre dos beacons se denomina superframe (supertrama) y oscila entre 15ms y 245s. El intervalo se divide en 16 slot temporales independientemente de la duración del mismo. Dentro de estos slots temporales, cualquier dispositivo puede transmitir siempre y cuando finalice su transmisión antes de la llegada del siguiente beacon y empleando para la contención del medio CSMA-CA (Carrier Sense Medium Access with Collision Avoidance). Con el fin de asegurar una baja latencia a ciertos dispositivos, el coordinador de la red puede asignar slots para el uso de esos dispositivos en concreto. Estos slots son los denominados GTS (Guaranteed Time Slots) y todos juntos forman el período libre de contención (CFP: Contention Free Period). Por otro lado, los slots no dedicados forman el período de acceso por contención (CAP: Contention Access Period). El CFP se sitúa justo antes del siguiente beacon y, en caso de que exista, aquellos dispositivos que transmitan durante el CAP deberán hacerlo garantizando que terminarán antes del comienzo del CFP. La información de la duración y propiedades de la supertrama viene determinada por el coordinador de la red a través del envío de los beacons. En la Figura 3-5 observamos cómo se distribuyen los distintos períodos de la supertrama.

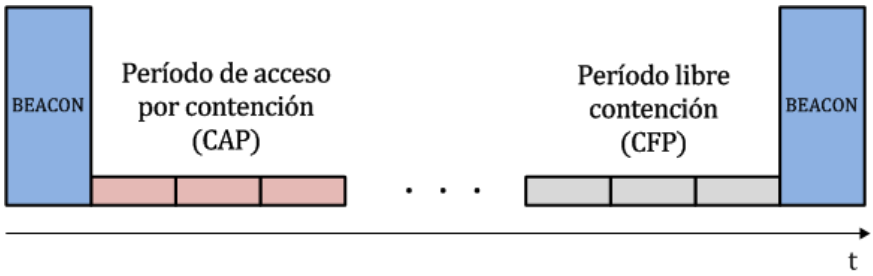

Figura 3-5: Estructura Superframe

- **Modo Beaconless (non-beacon-enabled network):** Los nodos pueden transmitir siempre y cuando el canal no esté ocupado. Por tanto, se emplea CSMA-CA para saber cuándo es posible acceder al medio. Cuando un nodo quiere transmitir espera un tiempo aleatorio (backoff), sí el canal está libre transmite, sino vuelve a esperar un tiempo aleatorio para volver a intentar la transmisión. Así hasta alcanzar un número máximo de repeticiones.

3.1.3.3 - Direccionamiento

El coordinador de la PAN asigna un identificador único a la red (PAN ID) para controlar la comunicación entre los nodos que pertenecen a su red y así poder distinguirlos de otros nodos. Además de un identificador de la PAN, cada dispositivo dispone de un identificador único a nivel de enlace de 64 bits (EUI-64[4]) para distinguirlo de lo demás nodos de la red. Sin embargo, resulta habitual emplear identificadores a nivel enlace de 16 bits (short MAC) dentro de la red, para reducir el tamaño de los campos de direccionamiento de la cabecera de la trama.

3.1.3.4 - Encaminamiento

El estándar IEEE 802.15.4 sólo define hasta el nivel enlace por lo tanto no soporta directamente la idea de encaminamiento, pero si se contempla el empleo de protocolos de nivel superior que puedan explotar esta técnica.

3.1.3.5 - Seguridad

IEEE 802.15.4 define tres niveles diferentes de seguridad[7]. Podemos clasificarlos de manera general en función de las propiedades que ofrecen: sólo encriptación (AES-CTR[5]), sólo autentificación (AES-CBC-MAC[6]), y encriptación con autentificación (AES-CCM[7]). Cada opción proporciona tres variantes en función del tamaño de las direcciones MAC (32, 64 ó 128) empleadas. Resulta evidente que pueden existir también aplicaciones que no requieran ningún tipo de seguridad, no siendo necesario implementar ninguno de estos métodos.

[4] Extended Unique Identifier
[5] Advanced Encryption Standard-Counter.
[6] Advanced Encryption Standard-Cipher block chaining with MAC.
[7] Advanced Encryption Standard-Counter with CBC-MAC.

3.2 - SWAN

SWAN (SAYME Wireless Area Network) [8] es el protocolo propietario de nivel aplicación empleado en la plataforma tecnológica SENSbee. Ha sido desarrollado íntegramente por la compañía SAYME Wireless Sensor Network.

Hereda las propiedades del estándar IEEE 802.15.4 para redes de sensores distribuidos de baja tasa binaria y bajo consumo.

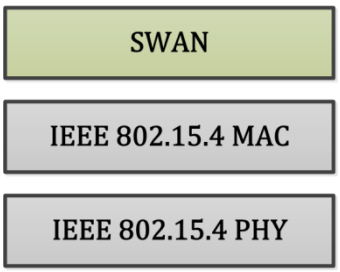

Figura 3-6: Stack de SWAN

3.2.1 - Características

3.2.1.1 - Tipos de dispositivos

Se definen dos tipos de dispositivos, SPOTs y MASTERs. Su funcionalidad está basada en los dispositivos RFD y FFD respectivamente, descritos en el estándar IEEE 802.15.4.

Los SPOTs se encargan de adquirir las señales de uno o varios de sus sensores, de su procesamiento y de la transmisión de los datos al MASTER, mediante el protocolo inalámbrico SWAN.

El MASTER por su parte, actúa siempre como coordinador de la red. Su función principal es recibir la información de los SPOTs y transmitirla mediante alguna de sus interfaces (WiFi, Bluetooth, RS232, etc.) a un servidor externo, para su almacenamiento y posterior utilización.

Los SPOTS funcionan mediante el uso de baterías, por lo que para asegurar una larga autonomía se implementan estrategias de bajo consumo. Son dispositivos que, debido a esta razón, se encuentran la mayor parte del tiempo dormidos (en torno al 99.9%). Durante este estado, los SPOTs desactivan varios módulos internos (transceptor, sensores, etc.) con el objetivo de lograr el menor consumo posible. Periódicamente, se despiertan para realizar sus tareas (adquisición y trasmisión) y cuando finalizan, vuelven otra vez a dormirse. Resulta fundamental implementar una estrategia que pueda gestionar todo este proceso.

Por otro lado, los MASTERs son dispositivos que siempre tienen que estar activos, ya que tienen la tarea de mantener el sincronismo de la red y gestionar las comunicaciones con los SPOTs. Por lo general, siempre estarán conectados a una fuente de alimentación constante.

La política de baja complejidad característica de las redes que operan con SWAN, hace que sus sistemas presenten una alta fiabilidad y requieran un mínimo mantenimiento. Por ello, implementan exclusivamente la topología en estrella. El MASTER actuará siempre como nodo coordinador de la red y los SPOTs como end-devices. Las comunicaciones sólo podrán tener lugar entre un SPOT y su MASTER, no existiendo ningún otro tipo de comunicación intermedia.

De esta forma, se consigue simplificar el proceso de comunicación y en gran medida el consumo energético, puesto que no se utilizan técnicas de encaminamiento dinámico que, como resulta lógico, requieren dispositivos con mayores recursos hardware y por lo tanto, un mayor consumo asociado.

3.2.1.2 - Modo Superframe

Debido al planteamiento de bajo consumo del protocolo SWAN, los SPOTs son dispositivos que pasan la mayor parte del tiempo en un estado de muy bajo consumo. La baja latencia de estos dispositivos en la red hace que sea interesante dedicar periodos temporales o slots para gestionar su funcionamiento. Para ello, se define una estructura temporal denominada supertrama (superframe).

El coordinador de la red (MASTER) asigna a cada SPOT un slot temporal dentro de la superframe donde asegura una transmisión sin colisiones. Sólo durante el tiempo que dura el slot, el SPOT puede llevar cabo sus tareas y las posibles retransmisiones. Cuando termina, el SPOT entra nuevamente en un modo de bajo consumo hasta que el proceso se repita de nuevo. De este modo, conseguimos implementar un mecanismo mediante el cual los SPOTs están muy poco tiempo en modo activo, consiguiendo así un gran ahorro de energía.

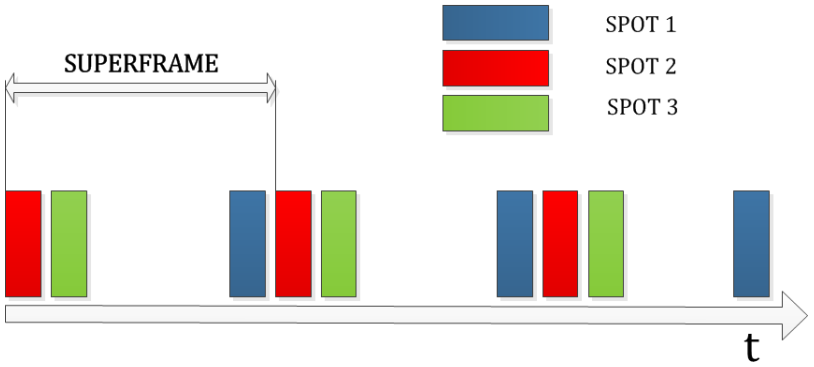

Figura 3-7: Sincronización de dispositivos

En la Figura 3-7, observamos un ejemplo de estructura superframe con tres SPOTs que tienen asignados un slot temporal cada uno. Con este modo de funcionamiento, además de conseguir una gestión del consumo de cada uno de los SPOTs muy eficiente, aseguramos también una transmisión ordenada y sin interrupciones. Sólo están activos durante su slot temporal, de esta manera evitamos por ejemplo que se produzcan transmisiones simultaneas. Por tanto, evitamos posibles retransmisiones.

Para llevar a cabo la sincronización temporal, el MASTER dispone de un contador (timer) que al llegar a un valor máximo se reinicia. El período de este contador se divide en intervalos o slots. Cada SPOT tiene asociado un slot único, en el cual deberá transmitir o perderá el turno hasta el período siguiente. Los SPOTs transmiten cuando su timer alcanza el valor máximo. Para conseguir que todos lo hagan en diferentes momentos (en su slot), se le aplica un retraso (offset) a sus contadores con respecto al del MASTER.

Por otro lado, el hecho de tener reservado un slot no significa que se utilice en todos los períodos. Pueden existir SPOTs que tengan su tiempo de muestreo mayor que el período de la superframe.

Resulta importante remarcar la idea de que la comunicación siempre es iniciada por los SPOTs en su slot. Además, SWAN implementa un algoritmo propio para la asignación de estos slots, realizando una distribución inteligente en el tiempo. Para ello, los slots son separados lo máximo

posible entre sí en el tiempo, de manera que los SPOT efectúan sus transmisiones los más distantes posible, evitando posibles colisiones entre ellos.

3.2.1.3 - Protocolo de comunicación inalámbrica

El protocolo de comunicación inalámbrica define las reglas a seguir durante la fase de transmisión de datos. Como se ha comentado anteriormente, el SPOT inicia siempre las transmisiones utilizando un algoritmo basado en el ahorro de energía. De esta manera, nada más despertarse el SPOT, comprueba si el canal está libre (CSMA-CA) y entonces transmite sus datos al MASTER que le contesta inmediatamente. De esta manera, se minimiza el tiempo que el SPOT debe estar despierto, al no tener que esperar a que sea el MASTER el que decida cuándo empezar la comunicación. Tal y como podemos observar en la Figura 3-8 existen dos modos de transmisión:

- **Transmisión simple:** Este es el caso más común, y consiste en el envío de una trama de datos del SPOT al MASTER. Este último responde que ha recibido correctamente los datos con una trama de confirmación (ACK).
- **Transmisión con configuración:** Se utiliza este modo de transmisión exclusivamente cuando es necesario realizar un cambio en la configuración de un SPOT. Para ello, el MASTER comprueba si existe alguna configuración pendiente y, sí es así, responde a la trama de datos que envía el SPOT con una trama ACK+Config, que confirma la recepción correcta de los datos y contiene además los nuevos parámetros de configuración del SPOT. Por su parte, el SPOT manda una trama ACK para comunicar al MASTER que se ha recibido correctamente la nueva configuración.

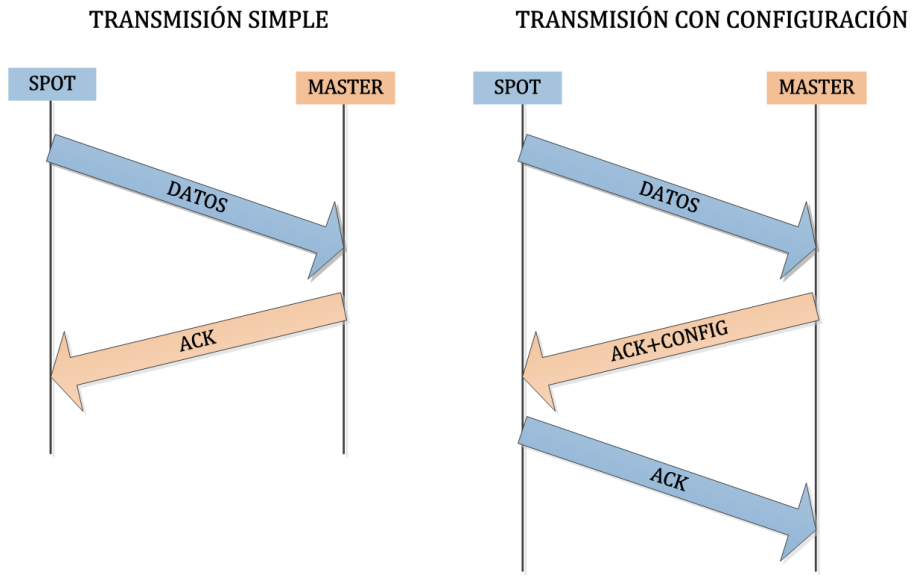

Figura 3-8: Modos de transmisión

El MASTER reconoce que el SPOT ha sido configurado correctamente cuando recibe un ACK. Sin embargo, para prevenir de posibles pérdidas de este ACK, que implicarían volver a enviar de nuevo la trama de ACK+Config, el MASTER espera a que el SPOT mande una nueva trama de datos. Comprueba si el ID de configuración del SPOT se corresponde con el valor que tiene guardado, y si el valor coincide, el MASTER asume que está correctamente configurado (el ID es único) y se vuelve al funcionamiento normal de la red.

Para aumentar la probabilidad de recepción de los datos se lleva a cabo una estrategia de retransmisiones. Si una trama enviada por el SPOT no es confirmada en un tiempo determinado (Time Out), se retransmite. Se establece un número máximo de retransmisiones, de modo que cuando se supera sin recibir ningún trama ACK, el SPOT se duerme (finaliza su slot temporal) y en

el período siguiente vuelve otra a vez a su funcionamiento normal (adquirir y transmitir la nueva información).

El protocolo SWAN establece tres tipos de tramas:

- **Tramas de datos** (Figura 3-9)**:** Siempre son enviadas por los SPOTs. Contienen un identificador del tipo de trama, tipo de magnitud y la propia medida. Además de un campo con configuración actual del SPOT. Si un sensor tiene todos los sensores desactivados enviará una trama vacía, sólo con el identificador de trama y el de configuración, para mantener la comunicación con el MASTER.

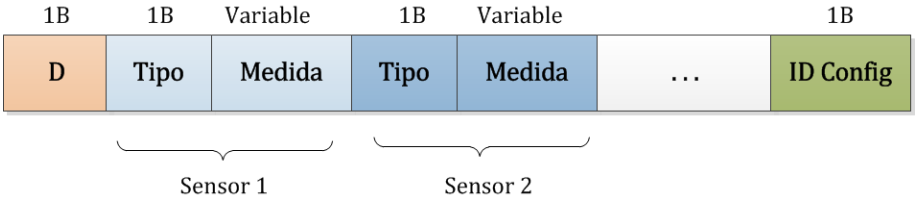

Figura 3-9: Trama de datos

- **Trama de confirmación** (Figura 3-10): Pueden ser enviadas por el SPOT o por el MASTER. Cuando es enviada por el MASTER contiene una referencia al temporizador (timer) que se utiliza para el sincronismo de la red. Sin embargo, cuando proviene del SPOT contiene el identificador de configuración actual.

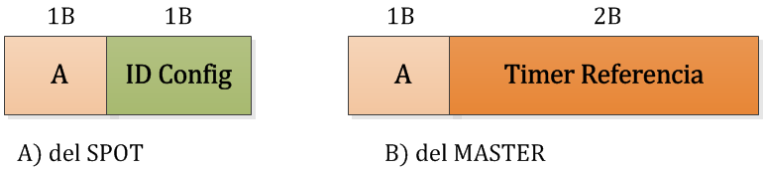

Figura 3-10: Trama de confirmación

- **Trama de confirmación más configuración** (Figura 3-11): Sólo puede ser enviada por el MASTER cuando hay una configuración pendiente de realizar en un SPOT. Sirve además como confirmación a una trama de datos. Contiene toda la información necesaria para configurar un SPOT: magnitud a medir, período de muestreo, número de canal, tipo de magnitud, módulo activado/desactivado, temporizador de sincronismo y el identificador de configuración.

Figura 3-11: Trama de ACK+Config

3.2.1.4 - Direccionamiento

El protocolo SWAN emplea el direccionamiento a nivel enlace que proporciona el IEEE 802.15.4. Las direcciones de los dispositivos emplean el formato corto (short) de 16 bits. Se dedican 8 bits al identificador de la red (PAN ID) y los 8 bits restantes al identificador del dispositivo (ID). Estas direcciones vienen predefinidas en el proceso de fabricación.

Aunque SWAN hace uso de un campo de direccionamiento suficientemente extenso como para gestionar un gran número de dispositivos, se ha demostrado experimentalmente que debido a las limitaciones hardware, un MASTER no puede gestionar un número mayor de veinte SPOTs.

3.2.1.5 - Encaminamiento

El protocolo SWAN propone únicamente el uso de la topología en estrella, donde todas las comunicaciones son siempre centralizadas. Al no existir ningún tipo de ruta alternativa no hay razón para implementar ningún tipo de técnica de encaminamiento.

Por otra parte, existen desarrollos no comercializados de SWAN que implementan repetidores [9] que permiten crear topologías en árbol. Aunque SWAN no soporta propiamente el encaminamiento a nivel enlace o red, si es posible gestionar de manera automatizada la selección de rutas desde el nivel de aplicación.

3.2.1.6 - Seguridad

El protocolo SWAN permite la aplicación de cualquiera de los métodos de seguridad a nivel enlace que proporciona el IEEE 802.15.4. Aunque no implemente por si mismo ningún protocolo de seguridad propio, las aplicaciones que explotan los datos de sus redes deben asegurar que otros usuarios no sean capaces de acceder a la información.

3.2.1.7 - Mantenimiento y gestión

No es un sistema auto-configurable. Debe ser diseñado, calculado y tendido en función de las necesidades de la aplicación en cuestión. Aunque una vez desplegado se pueden variar algunas características como el número de sensores activos y el período de muestreo de los SPOTS, mediante tramas de configuración.

Los SPOTs sólo conocen la dirección de su MASTER. Si por algún motivo sufriese alguna avería, se perdería la información que le transmiten los SPOTs hasta su remplazo. En ningún caso otro MASTER cercano se haría cargo de la situación, como sucede en otras tecnologías.

Si en cambio es un SPOT el que falla, a todos los demás dispositivos les pasaría inadvertida su pérdida; simplemente se dejarían de recibir sus datos. Puesto que el sistema utiliza una topología centralizada, no afectaría en nada a los demás dispositivos.

3.2.2 - Limitaciones

La robustez y la eficiencia de SWAN están basadas en su sencillez como protocolo de comunicaciones. Sin embargo, esto hace que muchas características útiles para entornos de redes de sensores no puedan ser planteadas.

Implementar el modo superframe para gestionar la eficiencia energética de los SPOTs y sus comunicaciones, implica que sólo pueda emplearse la topología en estrella. Por tanto, cuando surge un problema en la comunicación entre un SPOT y el MASTER, por ejemplo un obstáculo o una interferencia en el canal, la comunicación puede perderse tras un número de retransmisiones sin conseguir transmitir la información a su destino. Otros sistemas, por contra, implementan topologías distribuidas que permiten el uso de rutas alternativas para solventar este tipo de problemas.

El protocolo SWAN se caracteriza por funcionar con el uso de un hardware limitado, esto es así en gran parte porque se busca un muy bajo consumo en sus aplicaciones. Sin embargo, esto tiene repercusiones en el número de SPOTs que puede llegar a gestionar un MASTER, no pudiendo ser un número mayor de veinte. Resulta evidente que los sistemas distribuidos formados por una gran cantidad de dispositivos requieren de mayores recursos, por ejemplo para la gestión de las tablas de rutas empleadas para el encaminamiento.

En cuanto seguridad, SWAN implementa la protección proporcionada a nivel de enlace por el IEEE 802.15.4 y el que pueda tener la propia aplicación que explota los datos. Aunque pueda parecer suficiente, otros protocolos para redes de sensores dedican grandes esfuerzos a esta tarea. Implementan varios niveles de seguridad adicionales para prevenir posibles ataques.

Por último, es necesario comentar que los SPOTs no son dispositivos totalmente auto-configurables desde el punto de vista de su puesta en marcha en la red. Cuando son inicializados, requieren determinada información previa que no pueden obtener por sí mismos. Es necesario por ejemplo, introducirles la dirección MAC del MASTER que van a tener asignados, para que empiecen a funcionar como hemos descrito anteriormente. En otras tecnologías, cuando un end-device se inicializa en una red de sensores, no requiere ninguna información previa, ya que mediante el intercambio de mensajes de configuración con el nodo coordinador son capaces de empezar a funcionar de manera autónoma en la red.

3.2.3 - Implementación de SWAN

Para implementar el protocolo SWAN se emplea el microcontrolador MSP430F149. Pertenece a la familia de chips de ultra bajo consumo MSP430 de Texas Instruments (TI). Se caracteriza porque su arquitectura dispone de cinco modos de bajo consumo con el fin de optimizar la duración de las baterías [10].

El CC2420 es el transceptor RF que se usa como chip de comunicaciones para aplicaciones de bajo consumo. Constituye una solución de bajo costo y altamente integrado para una comunicación inalámbrica robusta.

3.3 - 6LoWPAN

Hoy en día resulta difícil dudar que Internet constituya uno de los mayores avances de la humanidad. Lo que empezó como una pequeña red de carácter académico, ha terminado por convertirse en la red que hoy todos conocemos. El Internet de los routers, servidores y hosts ha continuado evolucionando durante los últimos años hasta llegar a un nuevo escalón, el llamado *Internet of Things*.

Esta nueva visión tiene como objetivo extender el número de dispositivos conectados, haciendo posible que, por ejemplo, los objetos cotidianos puedan participar en la red. El impacto del *Internet of Things* será muy significativo en los próximos años proporcionando una mejora en la calidad de vida de la personas. Así pues, está considerado como el siguiente reto para la comunidad de Internet, usuarios y compañías.

Sin embargo, poder alcanzar una extrema capilaridad en la red requiere modificaciones en los protocolos que actualmente están en uso en Internet. Errores en la planificación original de la red han retrasado su crecimiento natural y el desarrollo de nuevas aplicaciones. Por tanto, conseguir que los dispositivos embebidos puedan conectarse a la red es fruto del desarrollo de nuevos estándares y protocolos que convierten a Internet en una red más global, dinámica y sin límite. El número de dispositivos embebidos con capacidad IP (Internet Protocol) está creciendo muy rápidamente y, aunque es difícil de estimar, seguramente superen el número de ordenadores personales y servidores en un futuro próximo.

3.3.1 - Definición 6LowPAN

6LoWPAN [11] es el acrónimo de *IPv6 over Low Power Wireless Personal Area Networks*. Se denomina también 6lowpan al grupo de trabajo en el área de Internet del IETF (Internet Engineering Task Force).

6LoWPAN proporciona mecanismos de encapsulación y compresión de cabeceras para la transmisión y recepción de paquetes IPv6 sobre enlaces IEEE 802.15.4. Las motivaciones del uso de IPv6 en redes IEEE 802.15.4 se recogen en el documento RFC 4919 y se resumen como:

- Las tecnologías basadas en IP ya existen, son bien conocidas y funcionan.
- Infraestructura de red existente.
- Es un estándar abierto.
- Herramientas para el diagnostico, gestión y comisionado de redes IP.
- Los dispositivos IP pueden conectarse a Internet (u a otras redes) sin la necesidad de entidades intermedias como gateways[8] o proxies[9].

Con el fin de aprovechar estas características, 6LoWPAN propone el uso de una capa intermedia de adaptación (Figura 3-12) que permita a los dispositivos de bajo consumo, baja tasa binaria y con un hardware muy limitado soportar de forma natural el protocolo IP.

Uno de los grandes retos que 6LoWPAN debe superar, es satisfacer el alto requisito que IPv6 impone al nivel de enlace en cuanto al tamaño mínimo de unidad máxima de transferencia (MTU: Maximun Transfer Unit). El estándar IPv6 específica que la capa de nivel enlace debe proporcionar un MTU de 1280 bytes. Esta cifra excede el tamaño máximo de la trama de enlace

[8] Un gateway es un dispositivo que permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación.
[9] Un proxy es un servidor que actúa de intermediario entre una conexión a Internet y una red interna.

IEEE 802.15.4, el cual es de 127 bytes. De modo que nos encontraríamos en un escenario en el que la capa de aplicación podría transportar una cantidad muy pequeña de información y, por tanto, el uso de IPv6 sobre IEEE 802.15.4 sería muy ineficiente en términos de energía y tiempo. Para hacer posible el cumplimiento del requerimiento de MTU mínimo, 6LoWPAN proporciona un mecanismo de fragmentación que permite descomponer los paquetes IPv6 en fragmentos más pequeños para que puedan ser transportados por la capa de enlace. Sin embargo, la fragmentación es un proceso costoso que en determinadas situaciones puede ser muy ineficiente. Con el fin de reducir su empleo, 6LoWPAN propone mecanismos de compresión para los paquetes IPv6.

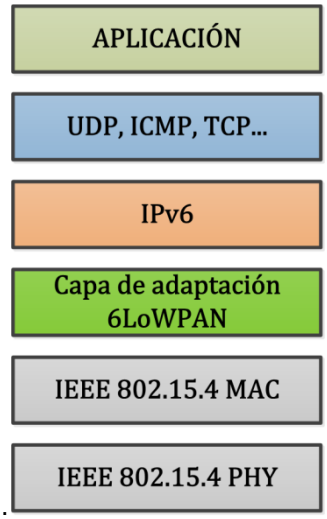

Figura 3-12: Stack de 6LoWPAN

Actualmente existen tres documentos RFC[10] del IETF que definen el funcionamiento y los objetivos del estándar 6LoWPAN. El RFC 4919 detalla una visión general del mismo, plantea las metas y los problemas a resolver en el futuro. Por su parte, el transporte de paquetes IPv6 sobre redes IEEE 802.15.4 aparece detallado en el RFC 4944. Los formatos de compresión de datagramas IPv6 sobre redes IEEE 802.15.4 pueden ser encontrados en el RFC 6282.

Título	Estado actual	Fecha modificación
Transmission of IPv6 Packets over Bluetooth Low Energy	ACTIVE	10-10-2011
Neighbour Discovery for Low Power and Lossy Networks	Publication Requested	24-10-2011
Problem Statement and Requirements for 6LoWPAN Routing	IESG Evaluation	07-02-2011
Design and Application Spaces for 6LoWPAN	IESG Evaluation	26-07-2011
Transmission IPv6 Packets over IEEE 802.15.4 Networks	RFC4944	04-04-2007
Compression Format for IPv6 Datagrams over IEEE 8021.54-Based Networks	RFC6282	24-02-2011
Overview, Assumptions, Problem Statement, and Goal.	RFC4919	02-03-2007

Tabla 3-2: Estado actual del estándar 6LoWPAN

[10] Un RFC (Request for Comments) es un documento cuyo contenido es una propuesta oficial para la creación y establecimiento de un estándar en Internet.

Como se puede apreciar en la Tabla 3-2, 6LoWPAN es todavía un estándar en vías de desarrollo. Por su parte, el IESG (Internet Engineering Steering Group) es el grupo perteneciente al IETF que se encarga de hacer las revisiones técnicas finales antes de su aprobación oficial. Algunos documentos están todavía pendientes de ser validados por él. No obstante, existen también otros grupos de trabajo que paralelamente se encargan de cuestiones específicas de 6LoWPAN como es, por ejemplo, ROLL (Routing Over Low Power and Lossy Networks). La tarea de dicho grupo consiste en el análisis de requisitos y estandarización de un protocolo de encaminamiento para aplicaciones embebidas.

3.3.2 - Características

3.3.2.1 - Compresión de cabeceras

En la Figura 3-13 se observa como el tamaño de las cabeceras impuesto por las capas de red y transporte supone un problema a la hora de implementar la tecnología IP en redes IEEE 802.15.4. En el caso más general:

- La cabecera de red IPv6 tiene un tamaño fijo de 40 bytes.
- La cabecera de transporte UDP (User Datagram Protocol) tiene un tamaño de 8 bytes.
- La cabecera de enlace IEEE 802.15.4 ocupa 23 bytes con direccionamiento EUI-64 (64 bits). Además, si implementamos AES-CCM-128 para la seguridad, el tamaño de la cabera aumenta hasta los 44 bytes.
- El campo FCS es de 2 bytes.

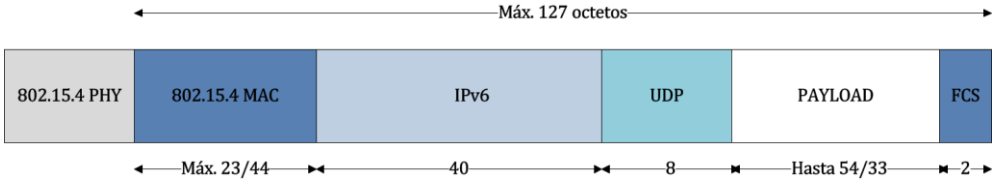

Figura 3-13: Paquete IPv6 en una red IEEE 802.15.4

Teniendo en cuenta que el tamaño de la trama a nivel enlace es de 127 bytes (incluyendo cabeceras), el payload disponible en función del uso de seguridad sería:

- **Sin seguridad**. 127-25(802.15.4)-40(IPv6)-8(UDP) = 54 bytes.
- **Con Seguridad AES-CCM-128.** 127-46(802.15.4)-40(IPv6)-8(UDP) = 33 bytes.

Resulta evidente que IPv6 no está optimizado para operar en redes IEEE 802.15.4. La cantidad de información de nivel aplicación que se podría transportar sería muy pequeña y resultaría altamente ineficiente debido a la sobrecarga producida por la transmisión de cabeceras. Por tanto, 6LoWPAN proporciona mecanismos de compresión que permiten reducir su tamaño.

En la Figura 3-14 se muestra cómo se realiza el proceso de compresión de la cabecera IPv6 a partir de las redundancias que contiene un paquete de red. Los campos que constituyen la cabecera, salvo el referente al número de saltos, pueden ser eliminados aprovechándonos de información proporcionada por la trama de enlace. Cuando se aplica 6LoWPAN sobre un paquete IPv6 es necesario añadir una cabecera de adaptación con información sobre el proceso de compresión. Como puede observarse en dicha figura, la compresión elimina aquellos campos de la cabecera IPv6 que no son estrictamente necesarios como: clase (class), flujo (flow) y siguiente cabecera (next header). Con los campos de direccionamiento IP sucede lo mismo puesto que dentro de la red pueden emplearse las direcciones MAC contenidas en la cabecera de enlace. Este mecanismo permite comprimir la cabecera IPv6 de 40 a 3 bytes en el mejor de los casos. Además, especifica también métodos adicionales para la compresión del protocolo de la capa de transporte UDP.

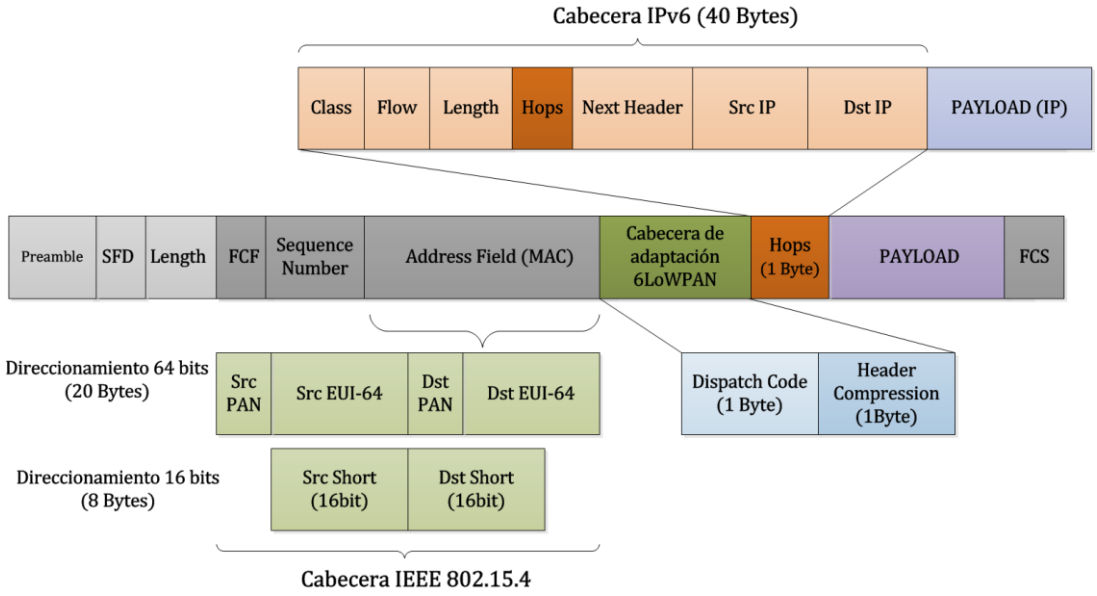

Figura 3-14: Compresión cabecera IP

En cuanto a los campos de la cabecera de adaptación, el Dispatch Code es un código numérico utilizado para identificar el tipo de trama. Nos permite saber, por ejemplo, si se ha producido compresión en la cabecera IPv6 o si existe fragmentación de paquete. En la Tabla 3-3 están representados algunos de sus valores más habituales:

Bit Pattern	Description
01 000001	Uncompressed IPv6 addresses
01 000010	HC1 Compressed IPv6 header
01 010000	BC0 Broadcast
01 111111	Additional Dispatch octe follows
01 XXXXXX	Mesh routing header
11 000XXX	Fragmentation header (first)
11 100XXX	Fragmentantion header (subsequent)

Tabla 3-3: Dispatch Code

La Figura 3-15 ilustra la cabecera de compresión (HC1: Header Compression) de la cabecera de adaptación. Proporciona información sobre que campos del paquete IPv6 han sido comprimidos y si existe compresión de otras cabeceras.

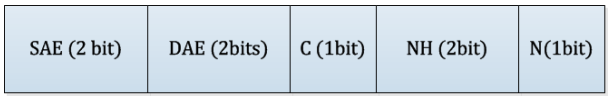

Figura 3-15: Header Compression 1 (Cabecera IP)

Los bits de Source Address Encoding (SAE) y Destination Address Encoding (DAE) indican que las direcciones IP han sido comprimidas, emplean un bit para prefijo y otro para el identificador del dispositivo; el bit C indica que los campos de clase de tráfico y control de flujo también han sido comprimidos; los bits que contiene NH informan sobre la cabecera que viene a continuación (00 desconocida, 01 UDP, 10 TCP, 11 ICMP); y por último el bit N indica si hay compresión en la siguiente cabecera (por defecto está a 0). Resulta habitual comprimir también la cabecera UDP para mejorar el payload disponible, para ello se añade otra cabecera de compresión a continuación de la cabecera HC1 con información sobre el proceso. Como se ilustra en la Figura 3-16, se indica mediante tres bits si se han comprimido los puertos origen/destino y longitud de la cabecera UDP. Los cinco bits restantes no tienen función actualmente.

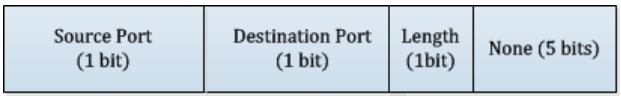

Figura 3-16: Header Compression 2 (Cabecera UDP)

Una de las ventajas que ofrece la compresión de cabeceras (HC1/HC2), además aumentar el payload, es que permite la comunicación entre dispositivos 6LoWPAN si que sean necesario otros requisitos adicionales.

Algunos de los casos más relevantes que nos podemos encontrar en función del nivel de compresión de las cabeceras son:

- **Cabeceras IPv6/UDP sin comprimir** (Figura 3-17)**:** El Dispatch code (01000001_2) indica que no existe ninguna compresión de cabeceras. El payload disponible puede variar entre 53/32 bytes en función del uso de seguridad.

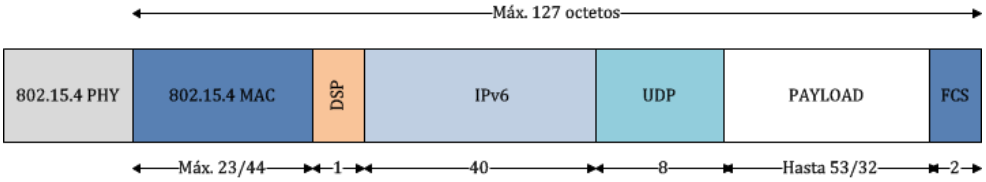

Figura 3-17: Peor caso

- **Compresión de cabeceras** (Figura 3-18): El Dispatch code (01000010_2) indica que se produce compresión en la cabecera IP. Además, es posible comprimir también la cabecera UDP (caso B) para aumentar el payload. Pueden aparecer cabeceras de otros protocolos sin comprimir después de los campos HC1 o HC2.

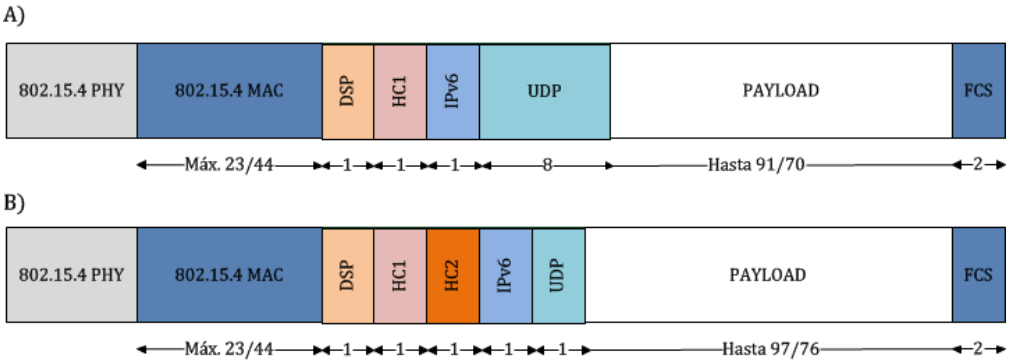

Figura 3-18: Mejores casos

La relación entre los bytes de información y los bytes totales es del 43% en el peor de los casos. Sin embargo, cuando implementamos la compresión 6LoWPAN es posible alcanzar una relación de 76%. En caso de emplear algoritmos de seguridad, aumenta el tamaño de las cabeceras como ya hemos comentado anteriormente, pero la diferencia entre ambos casos sigue siendo notable puesto que tendríamos una relación 26% en el peor de los casos frente a un 60% en el mejor de ellos. Por tanto, resulta evidente que el empleo de compresión aumenta considerablemente el payload disponible para el desarrollo de aplicaciones que empleen IPv6 sobre enlaces IEEE 802.15.4.

3.3.2.2 - Fragmentación

6LoWPAN proporciona un mecanismo de fragmentación y reensamblaje que permite descomponer los paquetes IPv6 en fragmentos más pequeños para que puedan ser transportados por la capa de enlace de manera transparente a IPv6. Se implementa mediante el uso de una cabecera adicional denominada Fragmentacion Header (4 bytes) para indicar el primer fragmento y otra, Subsequent Fragmentantion Header (5 bytes), para señalar los siguientes. La cabecera de fragmentación incluye el tamaño del datagrama y un identificador (tag) de paquete para distinguirlo de otros, los fragmentos restantes presentan adicionalmente un campo de offset en la cabecera para determinar su posición.

3.3.2.3 - Tipos de dispositivos

En la redes LoWPAN[11] (Figura 3-19), el edge router es el encargado de encaminar el tráfico dentro y fuera de la red. Integra dos interfaces distintas (IP y 6LoWPAN) para poder interactuar con dispositivos pertenecientes a distintos dominios y es, por tanto, el responsable del proceso de compresión de las cabeceras de red y transporte. Además, emplea una versión modificada del protocolo de descubrimiento de vecinos (ND: Neighbor Discovery) de IPv6 para la gestión y mantenimiento de la red.

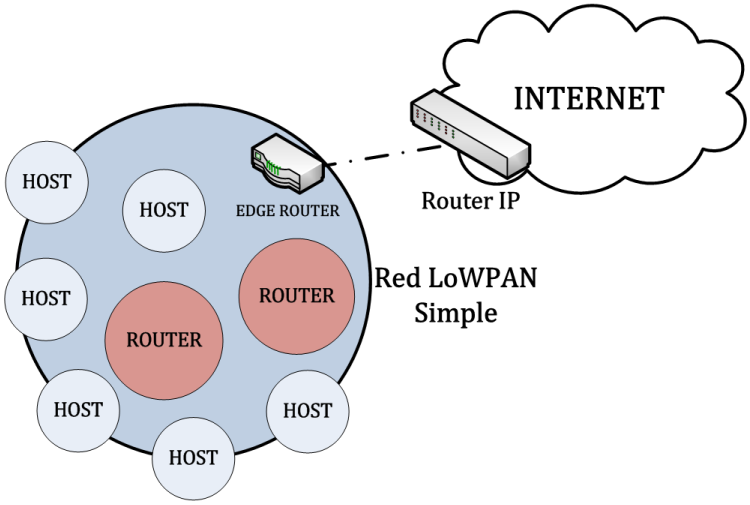

Figura 3-19: Red LoWPAN

Durante el proceso de inicialización, el edge router es el encargado de distribuir el prefijo que los nodos[12] emplean para generar sus direcciones de red (Figura 3-20). Para ello, responde a los RS (Router Solicitation) enviados por los nodos con RA (Router Advertisement). La respuesta del edge router, además de contener el prefijo, proporciona información adicional sobre la red. Una vez completado el proceso de autoconfiguración, todos los nodos de la red deberán registrarse en el edge router mediante el uso de los mensajes Neighbor Solicitation/Neighbor Advertisement (NS/NA). Con este proceso, el edge router lleva a cabo la detección de posibles direcciones duplicadas y almacena información sobre los nodos.

[11] LoWPAN es como se denomina a las redes que emplean 6LoWPAN.
[12] En 6LoWPAN es habitual emplear el término nodo para referirse a routers y hosts.

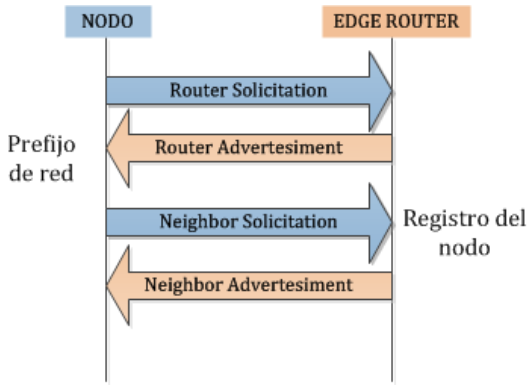

Figura 3-20: Inicialización de la red

Los routers son los únicos nodos con capacidad para el encaminamiento, esto supone que deben estar siempre activos en la red. Aprovechan los intercambios de mensajes RS/RA para configurar sus tablas de rutas y descubrir la topología de la red. En situaciones en las que un host no es capaz de alcanzar al edge router, los routers actúan como intermediarios en la comunicación y, también, colaboran en la propagación del prefijo de red. En determinadas implementaciones del estándar, los routers deben almacenar temporalmente la información destinada a los host, ya que es posible que éstos se encuentren dormidos.

Los hosts, en cambio, son los dispositivos más sencillos de la red. No requieren estar siempre activos como los routers por lo que es posible implementar en ellos estrategias de bajo consumo para el ahorro energético. No existe un método único para su gestión, cada una de los fabricantes que implementa la tecnología 6LoWPAN propone sus propios mecanismos. Por ejemplo, Texas Instruments [12] emplea el modo beacon característico del IEEE 802.15.4, similar al modo superframe empleado por el protocolo SWAN para la gestión de los ciclos de trabajo de los host. El edge router asigna a cada host un slot temporal donde asegura su transmisión sin colisiones. Sólo durante el tiempo que dura el slot, el host puede llevar a cabo sus tareas y las posibles retransmisiones. Cuando termina, el host entra nuevamente en un modo de bajo consumo hasta que el proceso se repita de nuevo. Por otro lado, Jennic [13] emplea el denominado modo encuesta (polling). Los host pueden dormir una gran parte del tiempo para conservar su energía de forma que cuando un dato es enviado al host desde otro nodo, es posible que no pueda ser recibido inmediatamente puesto que puede estar dormido. Por tanto, el router o edge router guarda los datos hasta que el host despierte y los solicite (polling). Es responsabilidad del host preguntar para comprobar sí existen datos pendientes de ser entregados.

3.3.2.4 - Direccionamiento

Las direcciones IPv6 de los nodos de la red tienen un tamaño fijo de 128 bits. Presentan un bloque de 64 bits con el prefijo de red y otro de 64 bits con el identificador de interfaz (IID: Interface ID).

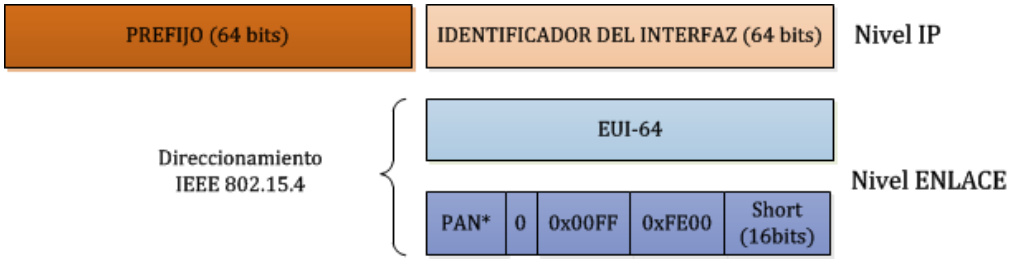

Figura 3-21: Direccionamiento 6LoWPAN

Durante el proceso de inicialización de los nodos en la red, el prefijo es distribuido por el edge router a todos mediante mensajes RA. Por su parte, el identificador de interfaz se obtiene directamente a partir de la dirección de enlace IEEE 802.15.4 (ver Figura 3-21), siendo posible emplear la versión completa EUI-64 o la corta de 16 bits. Cuando se emplea esta última, es necesario rellenar los 64 bits de identificador con campos adicionales y, en caso de no conocerse el identificador de la PAN, se completa con ceros.

Este método para la generación de la dirección de red se conoce como autoconfiguración sin estado (SSA: Stateless auto-configuration) y es una característica propia del protocolo IPv6.

Evidentemente, resulta habitual que muchas implementaciones de 6LoWPAN propongan utilizar el direccionamiento manual, sin autoconfiguración, para mejorar la eficiencia energética de la red. De esta manera, podemos evitar inundar la red de mensajes RA con información del prefijo. Por el contrario, algunas características interesantes de 6LoWPAN, como la movilidad entre redes o la inicialización automatizada de los nodos, no podrán ser llevadas a cabo sin el proceso de autoconfiguración.

3.3.2.5 - Arquitectura de una red 6LoWPAN

Una red LoWPAN es una colección de nodos 6LoWPAN que comparten el mismo prefijo de red. En la Figura 3-22 se ilustran las tres posibles arquitecturas de red:

- **LoWPAN Simple**: Red que dispone de un sólo edge router.
- **LoWPAN Extendida**: Red con varios edge routers, habitual en aplicaciones de movilidad o con un gran número de nodos. Pueden formar adicionalmente una subred con otros dispositivos IP.
- **LoWPAN ad-hoc**: Red que no está conectada a Internet. Uno de los routers se configura para que actué como un edge router.

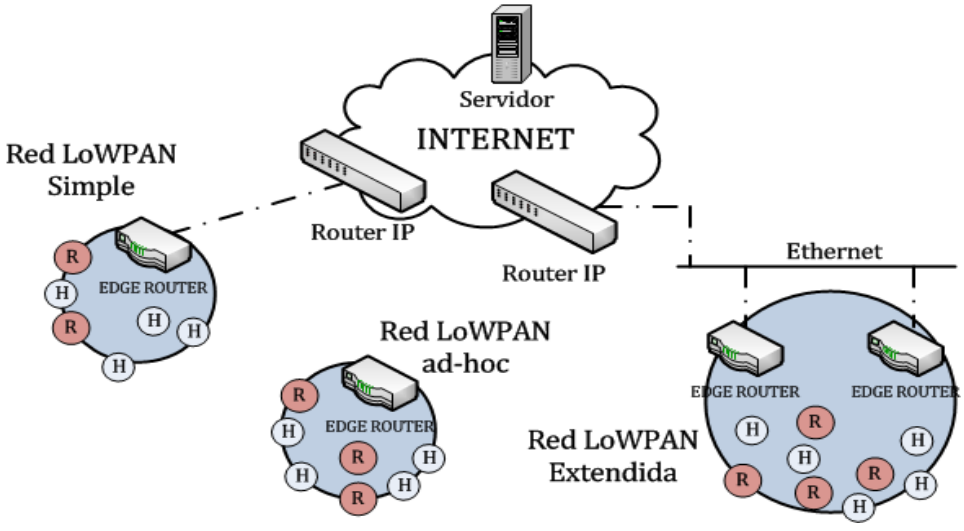

Figura 3-22: Arquitectura 6LoWPAN

3.3.2.6 - Encaminamiento

El encaminamiento en 6LoWPAN resulta complicado debido a que las capacidades de los nodos son muy limitadas en términos de energía, consumo y procesamiento. Existen numerosos protocolos que intentan solventar estos problemas con el objetivo de hacer del encaminamiento una técnica útil ante problemas como la aparición de obstáculos o la falta de alcance de los nodos. En función de en capa se lleve a cabo el proceso de encaminamiento, tenemos:

- **Mesh-under:** La capa de adaptación es la encargada de gestionar el proceso de encaminamiento. Este esquema emplea las direcciones de enlace EUI-64 (o la versión corta) para realizar las comunicaciones multisalto. Añade una cabecera adicional denominada cabecera de malla (mesh header) que contiene las direcciones MAC origen/destino y el número de saltos (por defecto 15). Cuando se utiliza mesh-under en 6LoWPAN, el nodo que envía el paquete escribe su dirección MAC en la cabecera mesh y la dirección MAC del destino. Al mismo tiempo, escribe su dirección y la del próximo salto en el campo de direccionamiento de la trama de enlace. Finalmente, el nodo origen transmite el paquete. Si el receptor es el destino del paquete, lo acepta, sino lo descarta. Por otro lado, reduce el valor del número de saltos y consulta el próximo salto en su tabla de rutas de enlace para actualizar los campos de la trama MAC, escribiendo su dirección en el campo origen y la del próximo salto en el destino. El paquete se descarta cuando el número de saltos es igual a cero o cuando alcance su destino final.
- **Route-Over:** Las decisiones de encaminamiento son tomadas por la capa de red, considerando cada salto a nivel de enlace como un salto a nivel IP. Los routers, como en cualquier red IP, utilizan tablas de rutas para determinar cuál es el próximo salto hacía el destino. Actualizan periódicamente la información de las tablas mediante el intercambio de información entre routers vecinos. Cuando se produce fragmentación en el paquete IP, los fragmentos son enviados al siguiente salto utilizando la información de la tabla de rutas. La capa de adaptación del siguiente salto comprueba los fragmentos recibidos y, si se han recibido correctamente, la capa de adaptación vuelve a juntarlos para enviarlos a la capa de red.

El encaminamiento mesh-under no resulta efectivo cuando se ven involucradas diferentes PANs puesto que no existe forma de avisar del cambio a los dispositivos. Además, en el caso de que se produzca fragmentación, la pérdida de uno de los fragmentos producirá la retransmisión completa del paquete IP desde el origen, siendo un proceso muy ineficiente en términos de energía y tiempo cada vez que tiene lugar. Por otro lado, el encaminamiento router-over emplea un alto consumo de ancho de banda y de energía debido a los dominios de broadcast de los enlaces IP. Además, en el caso de que produzca la pérdida de alguno de los fragmentos, se volverán a retransmitir todos desde el último salto, pero no desde el nodo origen como sucedía con mesh-under.

Por ello, el grupo de trabajo ROLL del IETF trabaja actualmente en el protocolo de encaminamiento RPL (Routing Protocol designed for Low Power and Lossy Networks) con el fin de solucionar las dudas que aún presenta el encaminamiento en 6LoWPAN.

3.3.2.7 - Seguridad

6LoWPAN propone diferentes mecanismos de seguridad en cada una de las capas que emplea. A nivel de la capa de enlace, como fue comentado en la sección relativa al IEEE 802.15.4, permite emplear mecanismos de encriptación basados en el cifrado de bloques AES (Advanced Encryption Standard). En la capa de red, se emplean los mecanismos definidos por el IPsec, Secure-ND (SEND), específico para el protocolo Neighbor Discovery, o Lightweight Secure Discovery for Low-power and Lossy Networks (LSEND), específico para 6LoWPAN-ND. En la capa de transporte, se propone emplear protocolos basados en UDP como Datagram Layer Security (DTLS), el cual es una adaptación del Transport Layer Security (TLS). Los mecanismos de seguridad de red y transporte citados, aún están en proceso en debate dentro del IETF.

3.3.2.8 - Mantenimiento y gestión

El protocolo Neighbor Discovery (ND) es un protocolo de nivel de red que emplea mensajes ICMPv6 (Internet Control Message Protocol for IPv6). Las funciones que desempeña en la red son: descubrimiento de routers, distribución de prefijo y parámetros de red, autoconfiguración de direcciones, resolución de direcciones de la capa de enlace (Address Resolution), determinación del próximo salto (Next-hop Determination), detección de direcciones duplicadas (DAD: Duplicate Address Detection), redirección (Redirect) y detección de vecinos inaccesibles (NUD: Neighbor Unreachability Detection). 6LoWPAN emplea una versión modificada denominada 6LoWPAN-ND debido a que algunas características del protocolo son inapropiadas para las redes inalámbricas de sensores que emplean IEEE 802.15.4.

El protocolo ND hace uso de mensajes multicast para llevar a cabo sus tareas, esta característica presenta serios inconvenientes debido a la naturaleza de las redes LoWPAN. El multicast no es compatible de forma nativa en el estándar IEEE 802.15.4 y, aún siendo posible, tiene asociado un elevado consumo energético puesto que obliga a los host a transmitir periódicamente información (muchas veces innecesaria), además de tener que recibir y procesar mensajes que probablemente descarten.

Por otro lado, ND asume equivocadamente que los hosts están siempre al alcance de cualquier nodo, como sucede en las redes IP, y que disponen de la capacidad de poder comunicarse con todos ellos (enlaces transitivos). Todas estas cuestiones obligan a tener que adaptar el protocolo.

La revisión 6LoWPAN-ND propone las siguientes los siguientes cambios:

- **Interacciones host-router iniciadas siempre por el host.** Permite de esta forma que el protocolo siga funcionando mientras los hosts pueden permanecer dormidos.
- **Registro de direcciones.** Permite llevar a cabo las funciones DAD, NUD y resolución de direcciones.
- **Eliminación de la resolución de direcciones basada en multicast**. Se emplea sólo con direcciones globales y se realiza a través del router mediante la funcionalidad del registro de direcciones. Reduce de esta manera el uso de multicast y el número de transmisiones entre host, produciendo un doble ahorro energético.
- **Eliminación de mensajes Redirect**. Asume que los enlaces son no transitivos.

3.3.3 - Implementación 6LoWPAN

6LoWPAN es una tecnología de red empleada en dispositivos embebidos. Como se ilustra en la Figura 3-23, disponemos de tres presentaciones habituales para su aplicación:

- **Solución Single-Chip (SoC)**: Dispositivos donde minimizar el coste y el tamaño es crítico. Se caracteriza por integrar las aplicaciones, el stack y el chip de comunicaciones en un sólo chip. TEXAS Instruments dispone dos modelos: TI CC2530 y TI CC1110. Por su parte, Jennic propone el modelo JN5139.
- **Solución Two-Chip**: En casos donde la aplicación es más compleja, se emplea esta solución donde un chip contiene el microcontrolador con las aplicaciones y el stack; otro se encarga de las comunicaciones. Mediante un puerto UART/SPI se comunican ambos chips. TEXAS Instruments dispone del modelo TI CC2520.
- **Solución Procesador de red**: En situaciones donde el diseño o la aplicación software ya existe. Se utiliza a parte un chip que integra el transceptor y también el stack. Suele emplearse para funcionar como router o edge router. Un ejemplo es el modelo CC1180 de Texas Instruments.

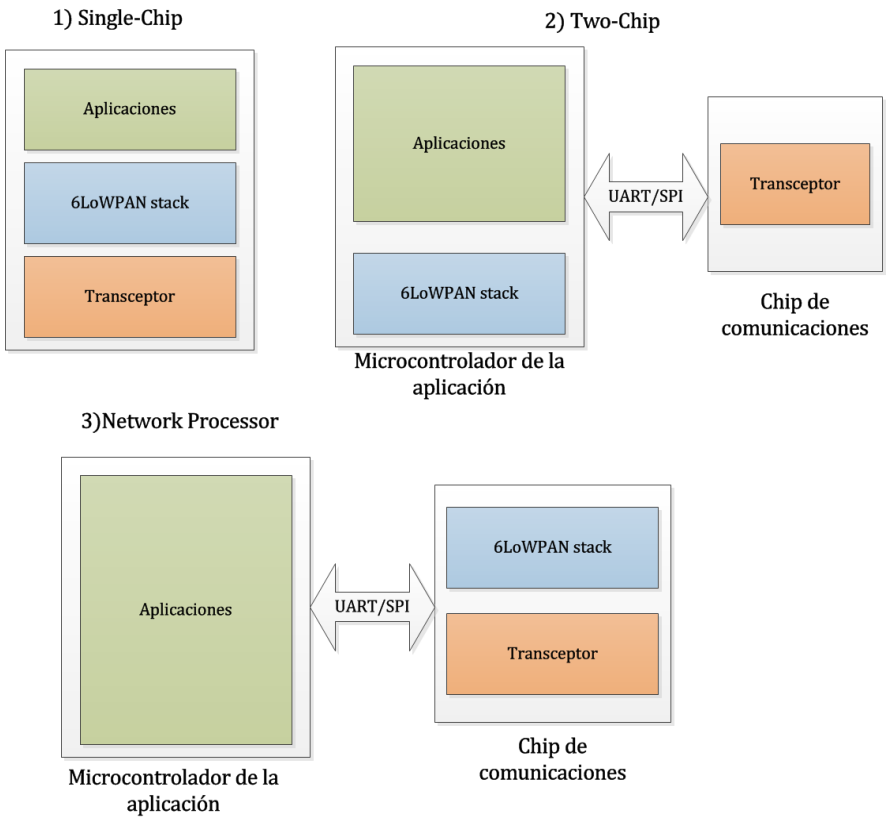

Figura 3-23: Implementaciones 6LoWPAN

3.4 - ZigBee

ZigBee es un estándar [14] desarrollado por la ZigBee Alliance que define un conjunto de protocolos de comunicación para el intercambio de datos inalámbricos entre dispositivos de bajo coste, bajo consumo y baja tasa de datos. Como se ilustra en la Figura 3-24, describe los niveles de red, seguridad y aplicación de las redes de sensores inalámbricas que emplean el nivel físico y de acceso al medio del estándar IEEE 802.15.4.

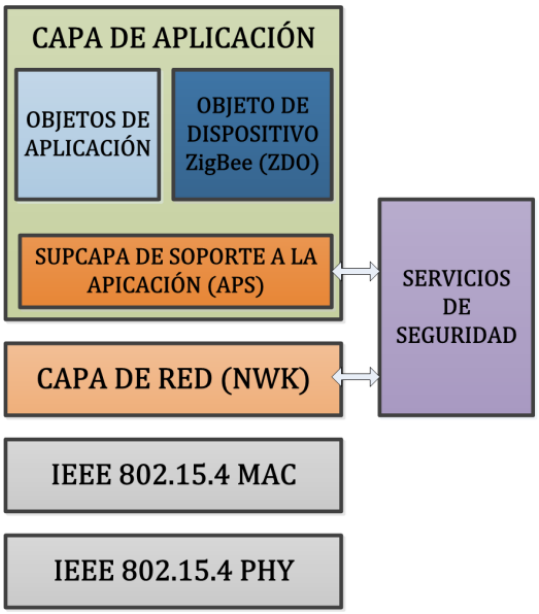

Figura 3-24: Stack de ZigBee

La ZigBee Alliance es un consorcio formado por más de 300 empresas de tecnología sin ánimo de lucro, entre las que se encuentran empresas como Philips, Samsung, Mitsubishi Electrics, Texas o Siemens, entre muchas otras. Su objetivo es crear una especificación abierta para definir topologías de malla y de árbol con aplicaciones interoperables en sistemas de control remoto. Por tanto, es muy importante que los dispositivos que implementen la especificación ZigBee sean capaces de interactuar entre ellos a pesar del fabricante original. La interoperabilidad es una de las claves del protocolo ZigBee frente a otros protocolos de redes de sensores inalámbricas. Para ello, la ZigBee Alliance se encarga de comprobar y certificar aquellos productos que emplean su protocolo. Además, organiza regularmente eventos con los fabricantes para probar y verificar que sus productos funcionan entre ellos.

Para lograr la compatibilidad entre productos, ZigBee define los denominados perfiles de aplicación (application profiles) en el nivel de aplicación. Los profiles describen cómo las aplicaciones (application objets) de distintos dispositivos pueden conectarse entre sí y cooperar para realizar una determinada tarea. De hecho, las redes ZigBee parten de la idea de estar constituidas por productos de diversos fabricantes. Por otro lado, la ZigBee Alliance ha definido dos versiones diferentes del stack en la última especificación (ZigBee-2007), ZigBee y ZigBee Pro, con el objetivo de promover aún más la interoperabilidad entre sus productos certificados. En función del nivel complejidad de nuestra red, será posible escoger entre una de las dos versiones.

3.4.1 - Tipos de dispositivos

Hay tres tipos de dispositivos en una red ZigBee: coordinador, router y end-device. Toda red ZigBee deberá contar con al menos un dispositivo que actúe como coordinador.

- **Coordinador (ZC):** Es el dispositivo encargado de la puesta en marcha de la red. Escanea su entorno para comprobar si existen otras redes, escoge el mejor canal para la comunicación y el identificador de red (PAN ID) para inicializar la red. Además, el nodo coordinador gestiona los mecanismos de seguridad de la red. Una vez finalizado el proceso de inicialización, la red no requiere de su presencia para seguir funcionando debido a la naturaleza distribuida de ZigBee.
- **Router (ZR):** Permiten que otros dispositivos puedan unirse a la red, extienden la cobertura de la red empleando técnicas de encaminamiento y ayudan a gestionar las comunicaciones con los end-devices que implementan estrategias de bajo consumo. Son dispositivos que requieren, por tanto, estar siempre alimentados.
- **End-Device (ZED):** Son los dispositivos más sencillos de la red, no soportan técnicas de encaminamiento y suelen emplear mecanismos para la gestión de su consumo.

ZigBee permite el empleo de las tres topologías de red definidas por el IEEE 802.15.4 (ver Figura 3-4):

- Estrella.
- Árbol.
- Malla (mesh).

Sin embargo, la topología más habitual en la especificación ZigBee es la topología de malla (mesh), denominada peer-to-peer en el estándar IEEE 802.15.4. En ella, cada dispositivo puede comunicarse directamente con cualquier otro dispositivo de la red. Requiere que un FFD actué como coordinador para inicializar la red y el proceso de comunicación. En las redes peer-to-peer, todos los dispositivos que participan en la retransmisión de los mensajes son siempre FFDs. No obstante, los dispositivos RFDs (end-devices) pueden formar parte de la red pero sólo pueden comunicarse con un dispositivo en particular (un coordinador o router) de la red. Esta topología hace uso del nivel de red de la especificación ZigBee debido a la necesidad de emplear técnicas de encaminamiento para la retransmisión de los mensajes.

3.4.2 - Capa de red

La capa de red (NWK) está definida por la especificación ZigBee. Gestiona las capas física y enlace del estándar IEEE 802.15.4 y proporciona una interfaz de servicio al nivel de aplicación. Es responsable de:

- Inicialización de una red.
- Procesos de asociación y abandono de una red.
- Direccionamiento. Sólo los coordinadores pueden asignar direcciones de red.
- Encaminamiento.
- Seguridad.

A continuación se procederá a describir el funcionamiento de estos mecanismos.

3.4.2.1 - Inicialización de una red

La inicialización de la red sólo puede ser llevada a cabo por un coordinador que no esté asociado a otra red. Para ello, el coordinador indica a su capa MAC que realice una detección de energía para un determinado número de canales o, en su defecto, para todos. De este modo, se determina cual es el mejor canal para la comunicación. Una vez finalizado este proceso, el coordinador realiza un escaneo activo del canal para detectar sí existen otras redes IEEE 802.15.4/ZigBee empleando el mismo identificador de PAN. En función de los resultados la capa de red establece el canal y el identificador de PAN que se va emplear en la red. Por otro lado, el coordinador empleará por defecto la dirección MAC y de red 0x0000.

3.4.2.2 - Asociación y abandono una red.

Las redes ZigBee permiten dos modos de funcionamiento de la red para la gestión de las comunicaciones: beacon y beaconless. En este contexto, el término beacon hace referencia a un modo de funcionamiento de la red donde el coordinador emplea periodos regulares de tiempo para la gestión de la comunicación ente los diferentes dispositivos que integran la red. En función del modo empleado, los procesos de asociación y de comunicación pueden variar.

En el **modo beaconless** (non-beacon-enabled network), el coordinador realiza un escaneado activo (scan active) para encontrar el canal con el menor número de redes activas. A continuación, establece los atributos MAC que requiere para poner en marcha la red: canal, PAN ID y dirección de red. Cuando un dispositivo quiera unirse a la red, realizará un escaneo activo del medio difundiendo peticiones broadcast en cada uno de los canales. Cuando el coordinador reciba una de estas peticiones, enviará al dispositivo una trama con los parámetros de la red para empezar el proceso de asociación. Como se aprecia en la Figura 3-25, la trama de petición de asociación enviada por el dispositivo contiene su dirección de enlace completa (64 bits) para que el coordinador la registre. Después, el dispositivo enviará una trama de petición de datos para comprobar sí existe alguna información pendiente para él antes de su asociación a la red. Por último, el coordinador enviará una trama de respuesta a la asociación con la dirección de red (16 bits) asignada al dispositivo. Una vez recibida esta respuesta, el dispositivo formará parte de la red. Todas las tramas del proceso de asociación son unicast (entre el coordinador y el dispositivo) y de nivel enlace. Por tanto, durante este proceso serán necesarias confirmaciones (ACKs) de nivel MAC que aseguren la correcta recepción de las tramas de asociación.

En este modo de gestión, es habitual que los end-devices encuesten (polling) periódicamente al coordinador para saber si tienen alguna información pendiente que recibir. Para ello, envían una trama de petición de datos de forma unicast al coordinador o router más cercano. De esta manera, es posible que los end-devices estén dormidos la mayor parte del tiempo sin necesidad de atender a mensajes del coordinador (o router). Cada vez que despierten preguntarán si hay alguna información pendiente para ellos antes de llevar a cabo sus tareas. Por ejemplo, para saber si existen datos pendientes o si se han producido cambios en la configuración de la red.

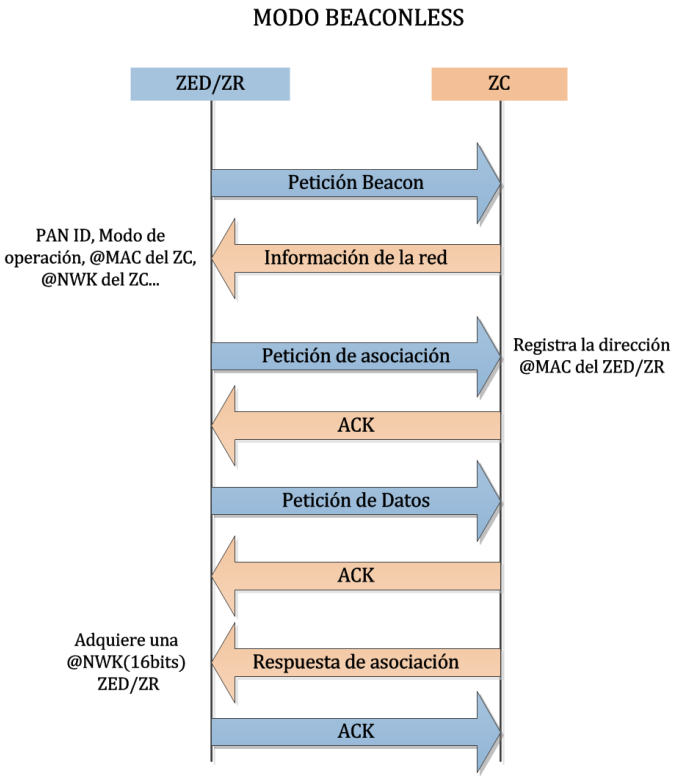

Figura 3-25: Modo Beaconless

Por otro lado, la especificación ZigBee permite también el empleo del **modo beacon** (beacon-enabled network) descrito en la sección del estándar IEEE 802.15.4. Sin embargo, no es muy habitual debido a la naturaleza de ZigBee, donde cada dispositivo puede transmitir o recibir información en cualquier momento sin necesidad de tener un intervalo de tiempo asignado para él.

3.4.2.3 - Direccionamiento

Los dispositivos ZigBee hacen uso de dos tipos de direcciones. Emplean una dirección única de 64 bits a nivel enlace (EUI-64 o dirección MAC) y una dirección de red de 16 bit (dirección lógica o de red). La dirección MAC es asignada por el fabricante durante el proceso de instalación. Esta dirección es única y global. Por otro lado, la dirección de red de 16 bit es asignada por el coordinador cuando el dispositivo se asocia a la red. Cuando se emplea la versión ZigBee del stack, la dirección de red refleja su posición en la red. Por ejemplo, la dirección 0x0001 reflejaría el primer ZR que se ha unido a la red. Sin embargo, en la versión ZigBee PRO las direcciones son asignadas de manera aleatoria.

3.4.2.4 - Encaminamiento

Para el encaminamiento de paquetes en la red, ZigBee emplea los siguientes métodos:

- Broadcasting.
- Encaminamiento en malla (mesh routing).
- Encaminamiento en árbol (tree routing). Sólo disponible en la versión ZigBee.
- Encaminamiento fuente (Source routing)**.** Sólo disponible en la versión ZigBee PRO.

Como puede observase en la Tabla 3-4, cada uno de estos métodos tiene sus ventajas y desventajas. La versión ZigBee PRO sustituye la técnica broadcast por multicast pero a efectos prácticos pueden considerarse equivalentes.

La técnica de broadcast consiste en que un nodo pueda alcanzar a otros nodos con una simple petición de datos. Este método no tiene asociado confirmación (el nodo original no tiene asegurado que el paquete llegue a su destino) para evitar que se produzcan saturaciones en la red. No se recomienda su empleo debido a que obligamos a todos los nodos a procesar mensajes que probablemente se desechen con el asociado consumo energético. Por tanto, la técnica de broadcast se considera un proceso de encaminamiento altamente ineficiente que debe intentar evitarse.

El encaminamiento en malla (mesh) emplea tablas con información para gestionar las diferentes rutas. Resulta muy eficiente en términos de tiempo, ancho de banda y recursos de memoria, una vez que la ruta se ha establecido. Los paquetes que se envían a través de la malla son confirmados, de modo que el nodo emisor puede saber si el paquete se ha recibido correctamente. Las rutas son distribuidas, lo que reduce el overhead de los paquetes que se envían. Una red ZigBee con topología de malla puede entregar paquetes superando hasta un total de 30 saltos.

El encaminamiento en árbol sólo está disponible en la versión ZigBee. Esta técnica emplea confirmación y es igual de eficiente en ancho de banda que el encaminamiento en malla, o incluso mejor en términos de memoria. Sin embargo, cuando un enlace se rompe, no hay posibilidad de recuperación. Por tanto, es habitual que ZigBee emplee encaminamiento en malla por defecto.

El encaminamiento fuente es exclusivo de ZigBee PRO y, como el encaminamiento en malla y árbol, requiere confirmación. Se utiliza cuando un coordinador (o router) necesita comunicarse con muchos nodos, quizás cientos o miles. Con la técnica de encaminamiento malla, cada ruta requiere una entrada en la tabla y los nodos ZigBee típicamente no tienen suficiente memoria como para mantener miles de rutas. En el encaminamiento fuente, un nodo simple puede tener almacenadas en la memoria todas las rutas. La ruta de una comunicación particular es mandada como parte del mensaje. No obstante, este proceso está limitado a un máximo de 5 saltos.

	Broadcast	**Mesh**	**Tree**	**Source**
Multi-hop	Hasta 30	Hasta 30	Hasta 10	Hasta 5 saltos
Múltiples destinos	SI	NO	NO	NO
Comunicación one-to-one	NO	SI	SI	SI
Eficiencia ancho de banda	NO	SI	SI	SI
Eficiencia payload	SI	SI	SI	NO
Confirmación (ACK)	NO	SI	SI	SI

Tabla 3-4: Características de las técnicas de encaminamiento

3.4.2.5 - Seguridad NWK

En el nivel de red, ZigBee utiliza encriptación y autentificación de paquetes. Como se observa en la Figura 3-26, mediante la encriptación se consigue que nodos ajenos a la red no puedan acceder al payload del paquete. ZigBee emplea autentificación de trama para prevenir que algún nodo malicioso inyecte paquetes en la red.

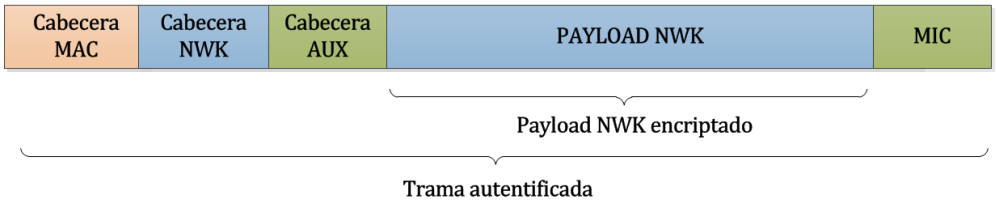

Figura 3-26: Seguridad NWK

Además, ZigBee empleará una llave de 128 bits en el nivel de red. Se denomina llave de red (network key). Se asume que, cuando un nodo conoce la llave de una red ZigBee, dicho nodo es considerado de confianza.

3.4.2.6 - Formato del paquete a nivel de red

Como se puede apreciar en la Figura 3-27, la cabecera del paquete de red (NWK) está formada por:

- **Frame Control**: Campo de 2 bytes que indica principalmente el tipo de paquete (comando o datos), versión del stack, información de encaminamiento y si se está empleando seguridad.
- **Destination/Source Address**: Direcciones de red (2 bytes cada una). Se emplea el valor 0xFFFF en el campo destino para enviar un paquete en modo broadcast.
- **Radio**: Variable empleada para contabilizar el número de saltos.
- **Sequence Number**: Con la dirección de origen y el número de secuencia es posible identificar cualquier paquete.
- **Destination/Source IEEE Address**: Direcciones de enlace (hasta 8 bytes cada una). Su empleo es opcional y viene indicado por el frame control.
- **Multicast Control**: Define los parámetros de la técnica de multicast. Su empleo es opcional y viene indicado por el frame control.
- **Source Route Subframe**: Incluye información sobre el encaminamiento fuente. Su empleo es opcional y viene indicado por el frame control.

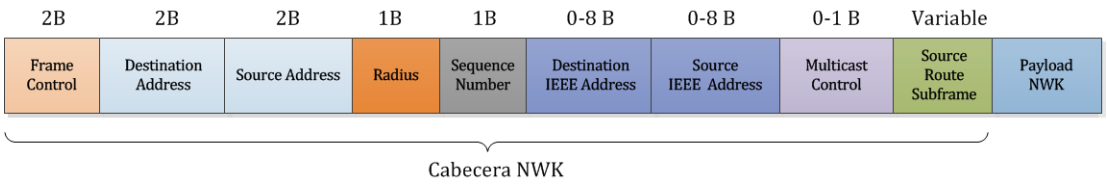

Figura 3-27: Paquete NWK

3.4.3 - Capa de aplicación

La capa de aplicación (APL) está constituida por tres secciones. Como se aprecia en la Figura 3-28: subcapa de soporte a la aplicación (APS: Application Support Sublayer), objeto de dispositivo ZigBee (ZDO: ZigBee Device Objects), y el marco de aplicación (AF: Application Framework).

La subcapa de soporte a la aplicación proporciona una interfaz entre la capa de red (NWK) y la capa de aplicación. Sus tareas habituales son:

- Mantener las tablas de vínculo (binding tables).
- Envío de mensajes entre dispositivos vinculados (bound devices).
- Gestionar grupos de direcciones.
- Mapear direcciones MAC de 64 bits en direcciones de red de 16 bits y viceversa.

ZDO es la aplicación que emplean las capas NWK y APS para implementar las funcionalidades de un dispositivo ZigBee:

- Define el rol del dispositivo en la red.
- Descubre dispositivos y aplicaciones de la red. Inicia o responde a peticiones de vínculo.
- Realiza tareas de seguridad.

El marco de aplicación en ZigBee es el entorno que contiene los objetos de aplicación (las aplicaciones) del dispositivo.

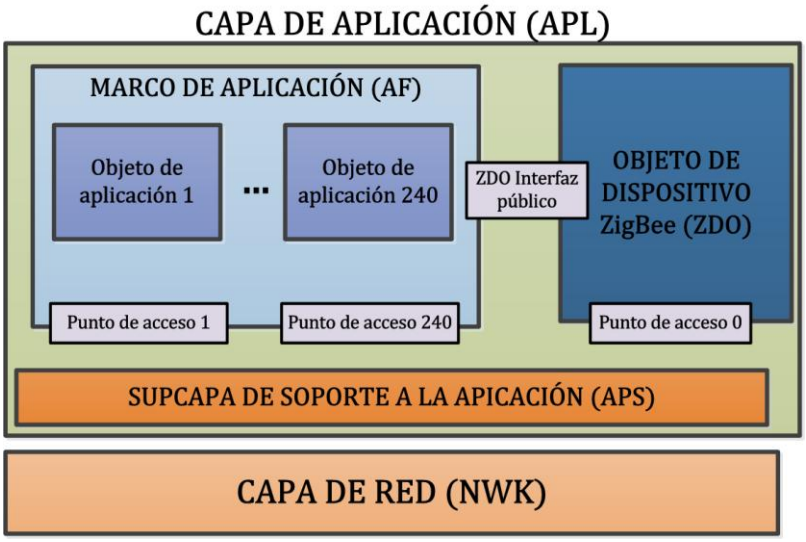

Figura 3-28: Capa de aplicación ZigBee

A continuación se detallarán las funciones de las subcapas que constituye el nivel de aplicación de ZigBee. Sin embargo, es necesario explicar antes una serie de conceptos que se emplean habitualmente en la capa de aplicación:

- **Perfil de aplicación (Application Profile):** Toda petición de datos en ZigBee es enviada y recibida a través de un perfil de aplicación. Un profile describe un dominio formado por un determinado tipo de aplicaciones y dispositivos. Existen profiles públicos especificados por la ZigBee Alliance y privados. Los profiles públicos son diseñados para que productos de un determinado fabricante puedan funcionar con productos de otro fabricante distinto.

- **Cluster:** Los clusters emplean un identificador de cluster (cluster ID) que se asocia al flujo de datos que entra o sale de un dispositivo. Definen el funcionamiento de una determinada aplicación mediante comandos y atributos. Con los clusters podemos, por ejemplo, encender una luz (comando) o ver sí está encendida (atributo). Sólo tienen significado dentro de un profile particular. Los enlaces se producen relacionando el identificador de cluster de salida con el identificador de cluster de entrada, siempre asumiendo que ambos clusters están en el mismo profile.
- **Puntos de acceso (Endpoints):** Son identificados con un número entre 1 y 240. Definen cada aplicación que funciona dentro de un nodo ZigBee. Además, permiten que aplicaciones de diferentes profiles convivan dentro de un mismo nodo.
- **Vinculación (Binding):** Proporciona un mecanismo para vincular uno o varios endpoints de diferentes nodos (ver Figura 3-29). Para ello, los nodos emplean una tabla de binding donde se almacenan las direcciones de enlace y red de los nodos vinculados. Esta tabla se actualiza automáticamente cuando se apunta una nueva dirección.

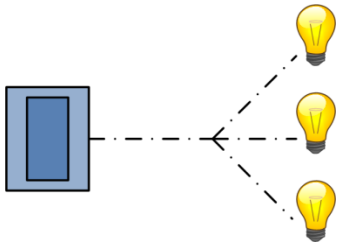

Figura 3-29: Binding

- **Dirección de grupo:** Un grupo es una colección de nodos que emplean una misma dirección. Una simple transmisión de datos puede ser recibida por todos los nodos de un grupo. El empleo de grupos es una característica opcional de la especificación de ZigBee, pero en determinados profiles, como en Home Automation, es obligatoria.
- **Descubrimiento de dispositivos:** Con este proceso, un dispositivo ZigBee puede descubrir otros dispositivos ZigBee mediante la realización de consultas, bien broadcast o bien, dirigidas a una determinada dirección. Existen dos formas de realizar estas consultas: con peticiones de dirección IEEE 802.15.4 o con peticiones de dirección de red (NWK). En el primer caso, las peticiones son unicast y se asume que la dirección de red es conocida. En el segundo caso, las peticiones son broadcast y llevan la dirección IEEE, que es conocida, en el campo de datos.
- **Descubrimiento de servicio:** Con este proceso, los servicios ofrecidos por los terminales (endpoints) de un dispositivo, pueden ser descubiertos por otros dispositivos de la red. Esta búsqueda puede lograrse realizando una consulta a cada endpoint de un determinado dispositivo o buscando características específicas de un servicio con consultas broadcast o unicast.
- **Tipos de mensajería**.
    - Direccionamiento directo: Una vez que los dispositivos se han asociado, los comandos pueden ser enviados de uno a otro. Los comandos son enviados al objeto de la aplicación correspondiente a la dirección destino. Por tanto, el binding no es un requisito para el direccionamiento directo. Además, se asume que con los procesos de descubrimiento de dispositivos y de servicios se han identificado un dispositivo y un endpoint que proporcionan un servicio complementario al del dispositivo que hace la petición. Así, los mensajes enviados directamente llevan la dirección completa del destino y toda la información del endpoint.
    - Direccionamiento indirecto: El uso del direccionamiento indirecto requiere el conocimiento previo de la dirección, del endpoint, del identificador de cluster y del identificador de atributo del dispositivo destino. Y además, es necesario que esa información esté en una tabla de binding en el coordinador antes de que se cree el mensaje con direccionamiento indirecto en el dispositivo origen.
    - Direccionamiento broadcast: Una aplicación puede enviar mensajes a todos los endpoints de un nodo destino. Este direccionamiento broadcast se produce a nivel aplicación. La dirección destino debe ser la dirección de broadcast de red de 16

bits y la fuente debe incluir en el mensaje el identificador de cluster, el identificador de profile y su endpoint.

Por tanto, la capa APS proporciona un entorno para las aplicaciones que controlan y gestionan las capas del protocolo ZigBee en un dispositivo. Los objetos de aplicación son desarrollados por los fabricantes. Un sólo dispositivo ZigBee puede contener numerosas aplicaciones (incluso con diferentes profiles) funcionando en él. Además, la posibilidad de emplear profiles en el desarrollo de aplicaciones permite extender la interoperabilidad entre productos desarrollados por diferentes fabricantes para una aplicación en concreto.

3.4.3.1 - Application Framework (AF)

Es el entorno encargado de gestionar las diferentes aplicaciones contenidas en el dispositivo. Cada aplicación dispone de un punto de acceso (endpoint) único. Los dispositivos ZigBee pueden disponer de hasta 240 aplicaciones diferentes. La dirección de endpoint 0 es exclusiva del ZDO y el 255 se emplea para enviar un mensaje broadcast a todas las aplicaciones de un dispositivo. Además, también gestiona los application profiles y clusters del dispositivo.

3.4.3.2 - ZigBee Device Objects (ZDO)

El ZDO es el responsable de la inicialización de la APS, NWK y del proveedor del servicio de seguridad (SSP: Security Service Provider). De manera similar a los application profiles definidos en la AF, existe un profile definido para el ZDO denominado perfil de dispositivo ZigBee (ZDP: ZigBee Device Profile) o device profile. El ZDP contiene los descriptores del dispositivo y clusters (sólo comandos). Además, este profile es soportado por todos los dispositivos de una red ZigBee. La función del ZDP es proporcionar soporte de descubrimiento de servicios y dispositivos, y el mecanismo de binding. El descubrimiento de dispositivos es una habilidad que permite determinar la identidad de otros dispositivos en la PAN. En el descubrimiento de servicios, el dispositivo puede pedir a otro dispositivo de la red que le proporcione información detallada sobre las aplicaciones que disponga o endpoints que tenga asignados.

3.4.3.3 - La subcapa APS

La subcapa APS proporciona un servicio de datos para las aplicaciones y para el ZDO. Además, gestiona la comunicación con la capa de red. Sus funciones más importantes son:

- Generación de la PDU de aplicación. La información de los objetos de aplicación y los mensajes de ZDO se encapsulan en su payload.
- Gestiona el intercambio de mensajes cuando se aplica binding entre dispositivos.
- Mejora la fiabilidad. Permite emplear mensajes de confirmación a nivel aplicación para asegurar la comunicación entre nodos cuando existen saltos entre ellos.
- Elimina mensajes duplicados.

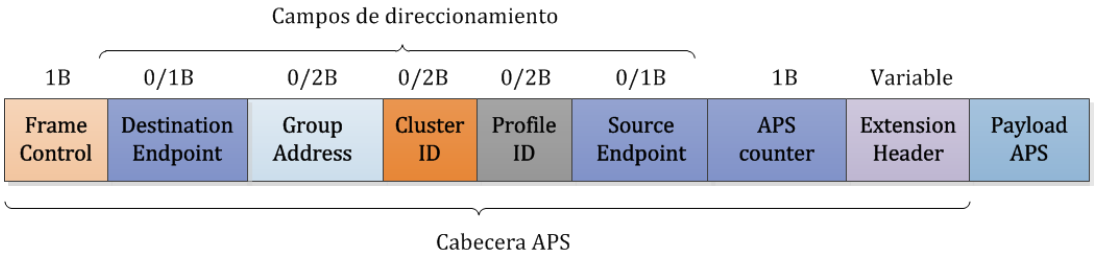

Figura 3-30: Formato paquete general APS

Como se puede observar en la Figura 3-27, la cabecera del paquete de APS está formado por los siguientes campos:

- **Frame Control**: Contiene información sobre el tipo de trama: datos, comando o confirmación; indica el modo de entrega: unicast, binding, grupo o broadcast; también informa del empleo de mecanismos de seguridad y de la petición de confirmación por parte del nodo emisor.
- **Destination/Source endpoint**: Direcciones origen/destino de los endpoints.
- **Group Address**: Dirección que referencia a grupo de endpoints destino, probablemente en distintos nodos. Cuando enviemos datos utilizando una dirección de grupo, todos los endpoints asociados a dicha dirección recibirán la trama.
- **Cluster ID**: Identificador de cluster.
- **Profile ID:** Identificador de perfil de aplicación.
- **APS counter:** Se emplea para evitar la duplicación de mensajes.
- **Extension header:** Permite futuras extensiones de la cabecera.

3.4.3.4 - Seguridad APS

Algunas aplicaciones requieren un nivel adicional de seguridad que el proporcionado por el nivel de enlace, por ejemplo, clientes que comparten una misma red pero donde cada uno puede tener datos confidenciales que no quiere que otros vean. Para ello, se emplea adicionalmente otra llave AES de 128 bits para proteger el el payload del paquete APS.

3.4.4 - Implementación de ZigBee

ZigBee es un protocolo de comunicaciones inalámbrico diseñado para chips de comunicaciones IEEE 802.15.4 y microcontroladores de 8 a 32 bits. Se denomina plataforma a la combinación del chip de comunicaciones, microcontrolador y firmware de ZigBee. Si la plataforma ha sido certificada por la ZigBee Alliance como *compliant* (no todas las plataformas lo son), entonces se denomina ZigBee Certified Platform (ZCP). El uso de plataformas certificadas permite a las OEMs[13] desarrollar un enorme abanico de productos ZigBee.

En la actualidad, numerosos fabricantes de chips ZigBee venden soluciones del tipo single chip (SoC) que integran en un mismo chip todos los componentes. Sin embargo, habitualmente el chip de comunicaciones también puede adquirirse de forma separada, permitiendo a los desarrolladores elegir el microcontrolador que van emplear. En la Tabla 3-5 están representadas la soluciones ZigBee ofrecidas por los fabricantes de chips más importantes.

	Pila de protocolo	Modelos	Tipo de solución
Freescale	BeeStack	-MC1312 -MC1313 -MC13324V -MC13201 -MC13202	-Single Chip -Single Chip -Single Chip -RF -RF
Texas Instruments	Z-Stack	-CC2530 ZNP -CC2531 -CC2530 -CC2520	-Procesador de red -Single Chip -Single Chip -RF
Jennic	Jennic	-JN5148	-Single Chip
Ember	EmberZnet	-EM351 -EM357	-Single Chip -Single Chip

Tabla 3-5: Soluciones ZigBee

Por otra parte, la mayoría de estos fabricantes ofrecen kits de desarrollo para la creación y evaluación de aplicaciones como son, por ejemplo, el kit de desarrollo CC2530 (TEXAS) o el JN5148 (Jennic). Estos kits proporcionan numerosos ejemplos de aplicaciones para facilitar al desarrollador el aprendizaje. De forma adicional, la mayoría de fabricantes de chips ZigBee suministran con sus productos la pila de protocolos (IEEE 802.15.4 y ZigBee).

Para el desarrollo del proyecto se eligió la pila Z-Stack de Texas Instruments. Su elección vino determinada porque su plataforma de desarrollo emplea las mismas familias de microcontrolador (MSP430) y de chip de comunicaciones (CC2520) que las utilizadas en la plataforma SENSBee de SAYME. Además, su descarga es gratuita e incluye un extenso manual con el que poder adaptar el stack de TEXAS a nuestra plataforma cómo se explicará en los siguientes capítulos.

[13] OEM (Original Equipment Manufacturer): compañía que adquiere productos o servicios de su fabricante original al por mayor y lo anexa a su producto o servicio propio para así venderlo. Por ejemplo, un fabricante X que adquiere microcontroladores de Texas Instruments para incluirlo en su hardware, está actuando como OEM.

3.5 - Conclusiones

En este apartado se analizan las características generales de los protocolos SWAN, 6LoWPAN y ZigBee, con el fin de determinar cuál de ellos resulta más adecuado para implementar una red de sensores comercial empleando, para ello, el mismo hardware que SAYME utiliza en su plataforma SENSBee.

Como punto de partida, todos los protocolos estudiados en este capítulo basan su funcionamiento en el empleo del IEEE 802.15.4 para el nivel físico y de enlace.

Por su parte, SWAN es un protocolo propietario ideado desde el punto de vista del consumo energético y de la sencillez, tanto hardware como software. Implementa un nivel aplicación con el que gestiona de manera simple parte de las características del estándar IEEE 802.15.4. En SWAN existen sólo dos tipos de dispositivos, SPOTs y MASTERs, basados en las definiciones de FFD y RFD especificadas por el IEEE 802.15.4. Adicionalmente, se definió un repetidor intermedio para aumentar el rango de cobertura entre un SPOT y su MASTER, pero debido a la política interna de SWAN en cuanto complejidad se descartó su empleo. Por otro lado, SWAN emplea siempre el modo superframe con una topología en estrella. Es decir, todas las comunicaciones son centralizas y los SPOTs sólo puede transmitir en intervalos de tiempo asignados por el MASTER. Por tanto, es posible conseguir una gestión muy precisa del consumo energético en las comunicaciones. Además, puesto que sólo emplea un nivel adicional en la pila OSI, las tramas IEEE 802.15.4 que encapsulan los paquetes de aplicación SWAN no contienen apenas overhead, simplificando su procesamiento en los nodos. Todas estas razones expuestas hacen de SWAN un protocolo de muy baja complejidad, que basa su política de funcionamiento en la sencillez y en el bajo consumo. Sin embargo, esta sencillez hace que los nodos que integran las redes SWAN requieran de una puesta en marcha y de una gestión manual, a diferencia de otros protocolos que implementan métodos de autoconfiguración. En cuanto a la seguridad, SWAN delega esta tarea en el IEEE 802.15.4 aunque el nivel de aplicación pueda hacer uso de llaves de seguridad adicionales.

A continuación se analizan los protocolos planteados como posibles sustitutos al protocolo SWAN para el diseño y desarrollo de una red de sensores inalámbrica. Los protocolos estudiados y analizados en el desarrollo del proyecto fin de carrera fueron 6LoWPAN y ZigBee.

6LoWPAN no es exactamente un protocolo de comunicación como pueden ser SWAN o ZigBee. Su función es definir una capa de adaptación para la capa IP que permite implementar la tecnología IPv6 sobre enlaces IEEE 802.15.4. En las redes LoWPAN disponemos de tres tipos de dispositivos: edge router, router y host. El edge router actúa siempre como coordinador de la red (FFD), además implementa los mecanismos de compresión y fragmentación necesarios para poder emplear paquetes IP en una red de sensores que emplee como nivel físico y de enlace IEEE 802.15.4. Por otro lado, los host actúan siempre como los dispositivos RFDs. A diferencia de SWAN, 6LoWPAN añade la figura del router (FFD). Este dispositivo tiene la capacidad de poder encaminar paquetes dentro de la red y de facilitar la comunicación entre los dispositivos de la red. Actualmente, 6LoWPAN no tiene un modo de funcionamiento específico, algunos fabricantes han optado por emplear el modo beacon (TEXAS) y otros el beaconless (JENNIC). Además, tampoco están definidos mecanismos que aseguren un bajo consumo en la red. Algunos protocolos IPv6 empleados en 6LoWPAN no han sido aún adaptados a las características limitadas de las redes IEEE 802.15.4, lo cual hace que en ocasiones el estándar pueda resultar muy ineficiente. En cuanto al mantenimiento y gestión, 6LoWPAN proporciona todas las herramientas disponibles por IP, en especial el protocolo Neighbor Discovery. Para la seguridad, 6LoWPAN proporciona cuatro niveles: IEEE 802.15.4, IP (IPSec), UDP (DTLS) y aplicación.

Por tanto, una de las grandes ventajas que proporciona 6LoWPAN respecto a otros protocolos es que permite emplear todas las herramientas habituales de las redes IP. Por ello, es posible concluir que está orientado a dispositivos más complejos que los que habitualmente se utilizan en las redes de sensores que, como comentábamos al inicio del proyecto, se caracterizan por: bajo coste, sencillez software/hardware y bajo consumo. En cambio, 6LoWPAN busca explotar directamente las poderosas funcionalidades que ofrece la red: TCP/IP, WebServices, XML, etc.

De modo que se hace necesario utilizar un hardware más potente que el propuesto por otras soluciones como SWAN o ZigBee. No obstante, aún es un estándar en vías de desarrollo por lo que es difícil emplearlo fuera de ámbitos educativos o experimentales. Todas estas razones hacen que sea imposible plantearnos utilizar 6LoWPAN como un firme candidato para sustituir al protocolo SWAN.

Por otro lado, ZigBee es un protocolo de comunicaciones que aporta el nivel de red y aplicación al estándar IEEE 802.15.4. Sus productos son certificados por la ZigBee Alliance, ofreciendo siempre la seguridad de un funcionamiento óptimo. Además, está específicamente diseñado para dispositivos de bajo consumo, bajo coste y baja tasa binaria. En cuanto a sus características, implementa tres tipos de dispositivos: ZC, ZR y ZED. Su definición es muy similar a la de los dispositivos descritos por el estándar IEEE 802.15.4, salvo por que añade el ZR. Este dispositivo actúa como router, permitiendo emplear técnicas de encaminamiento para aumentar la robustez y cobertura de la red.

ZigBee implementa los dos modos de funcionamiento del IEEE 802.15.4, beacon y beaconless. En función de cual empleemos, dispondremos de unas ventajas u otras, es decir, habrá aplicaciones donde será más conveniente emplear una red con sincronismo frente a otras donde puede ser todo lo contrario. No obstante, habitualmente se emplea el método beaconless puesto que permite explotar todas las funcionalidades de red que proporciona la especificación ZigBee. Al igual que 6LoWPAN, es un protocolo auto-configurable y por tanto, los nodos pueden inicializarse automáticamente en la red sin necesidad de configuración manual. Incluye la herramienta de gestión ZDO para la configuración y mantenimiento de la red. Por otro lado, aporta seguridad a nivel enlace, red y aplicación.

A diferencia de los otros protocolos, ZigBee posee un fuerte interés por la interoperabilidad entre fabricantes, de modo que las redes ZigBee están preparadas para que cualquier empresa pueda añadir sus productos sin problemas de compatibilidad a una determinada red. En cierta manera, esto también es cierto para el caso 6LoWPAN puesto que no existe red más heterogénea que Internet. Pero a día de hoy, las redes 6LoWPAN son terriblemente incompatibles entre sí, no es posible añadir nuevos nodos que implementen versiones del stack de diferentes fabricantes. Sin duda, la interoperabilidad y las certificaciones de la ZigBee Alliance, proporcionan a ZigBee un grandísimo valor añadido frente a 6LoWPAN a la hora de comercializar un producto.

Como conclusión podemos afirmar que ZigBee se presenta como una perfecta actualización de SWAN. Una de las mayores ventajas que tiene utilizar este protocolo es que podemos emplear el mismo hardware que el utilizado por la plataforma SENSBee, sin apenas realizar modificaciones. Por tanto, no se requiere de nuevas inversiones para llevar a cabo esta actualización, sino que simplemente bastaría con adaptar el hardware actual.

Emplear ZigBee nos aportará nuevas funcionalidades de autoconfiguración y de gestión de red: encaminamiento, descubrimiento de rutas, multisalto, etc. Además, como explicaremos más adelante, es un protocolo parametrizable que permite su adaptación a entornos muy concretos de aplicación de manera muy eficiente. Por último, debemos recordar el gran valor añadido que otorga utilizar el protocolo ZigBee en cuestiones de certificación de aplicaciones. La certificación de productos ZigBee asegura al cliente el correcto funcionamiento y la certeza de que es interoperable con otros productos de la marca ZigBee, pudiendo elegir entre el gran número de compañías que constituyen la ZigBee Alliance.

En el capítulo 5, analizaremos el hardware empleado y expondremos que modificaciones son necesarias para hacer funcionar el protocolo ZigBee en las placas propietarias que proporcionó SAYME para la elaboración del presente proyecto fin de carrera.

En el capítulo 6, mostraremos las modificaciones necesarias en las libreras HAL para hacer funcionar el stack de Texas Instruments y estudiaremos diferentes escenarios de aplicación, que mostrarán algunas de las funcionalidades más importantes del protocolo.

Capítulo 4: DESCRIPCIÓN HARDWARE

En este capítulo se llevará a cabo una explicación detallada de los principales componentes hardware que forman el sistema: el microcontrolador y el chip de comunicaciones. Además, se detallarán todas las modificaciones realizadas en el hardware para integrar el protocolo ZigBee en la placa propietaria de SAYME. Por último, se analizarán los componentes adicionales que proporciona la placa.

4.1 - Introducción

Uno de los principales objetivos de este proyecto era reutilizar el hardware de los módulos de SAYME ya existentes para implementar en ellos el protocolo de comunicación ZigBee. La idea era poder desarrollar una aplicación de prueba que empleará ZigBee en la misma placa en la que se estaba utilizando el protocolo SWAN.

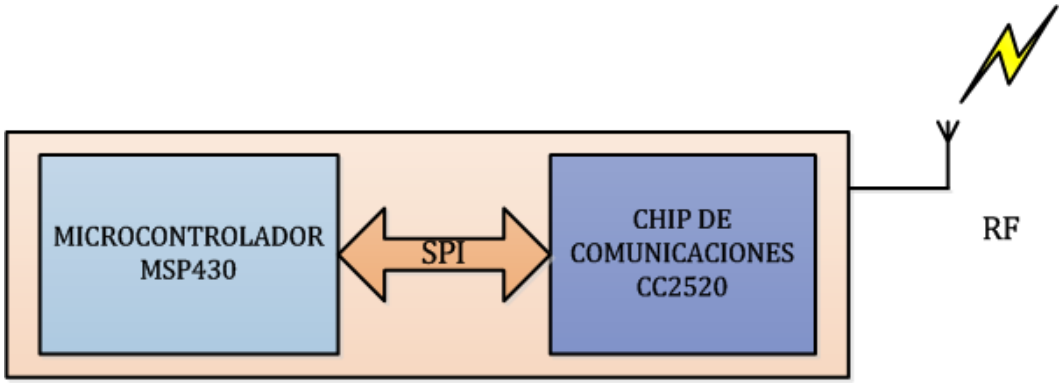

Figura 4-1: Esquema hardware de un módulo

La placa utilizada cuenta con un microcontrolador y un chip de comunicaciones, unidos entre sí mediante una interfaz puerto serie síncrono (SPI: Serial Peripheral Interface Bus). El microcontrolador es el que gestiona todas las operaciones del módulo: decide que acción hay que realizar en cada momento, almacena la información de sensores (si los hubiera) y de otros dispositivos, etc. Por su parte, el chip de comunicaciones es el encargado de llevar a cabo la comunicación inalámbrica y su funcionamiento es controlado desde el microcontrolador.

Adicionalmente, los módulos pueden disponer de sensores externos para realizar, por ejemplo, medidas de temperatura, presión, humedad, etc. Para ello, los sensores del módulo se conectan a través de un acondicionador de señal que adecua la salida de estos a la entrada del microcontrolador, pero no fueron empleados durante el proyecto ya que, como comentábamos anteriormente, nuestro objetivo principal era conseguir hacer funcionar el protocolo ZigBee en una placa ideada originalmente para SWAN y probar en ella una sencilla aplicación de muestra que verificase la correcta transmisión y recepción de información.

Estos componentes y otros de relevancia para la realización del proyecto se explicarán en las siguientes secciones.

4.2 - Descripción de los componentes

Los dos principales componentes hardware empleados son el microcontrolador y el chip de comunicaciones. El microcontrolador empleado pertenece a la familia MSP430, mientras que el chip de comunicaciones elegido ha sido el CC2520.

4.2.1 - Microcontrolador MSP430

Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, memoria (generalmente pequeña) y unidades entrada/salida. Aunque sus prestaciones son limitadas, sus principales características son su alto nivel de especialización y su reducido tamaño, llegando a ser, incluso, más pequeños que una simple moneda.

Su diseño busca el menor coste económico, por otra parte el consumo de un sistema en particular y su tamaño dependerá de la aplicación. Los recursos de un microcontrolador son muy limitados en comparación a los de un microprocesador, pero a diferencia de estos, los microcontroladores requieren de una mínima electrónica externa (cristales, condensadores, etc.) para funcionar.

No obstante, el microcontrolador dispondrá de un generador de reloj integrado y una pequeña cantidad de memoria, por lo que para hacerlo funcionar todo lo que se necesita son una serie de programas de control, alimentación y una mínima electrónica asociada. A esto podemos añadir las interfaces de entrada/salida, como ADC (Analog-to-Digital Converter), USCI (Universal Serial Communication Interface) o temporizadores, que nos permiten comunicarnos con dispositivos externos.

En nuestros sistemas, el microcontrolador nos permite controlar los distintos módulos y gestionar las tareas que necesitamos. Están específicamente diseñados para controlar equipos electrónicos de forma autónoma y su consumo y tamaño los hacen muy apropiados para el desarrollo de módulos integrados.

El primer diseño de la placa de SAYME para ZigBee utilizaba el microcontrolador MSP430F1419 de Texas Instruments. Se comprobó que sus recursos de memoria (60KB de Flash y 2KB de RAM) no eran suficientes para soportar el stack de ZigBee. Por tanto, se propuso utilizar el modelo de microcontrolador MS430F2618 [10] de la misma familia. Este chip proporciona más memoria (RAM y FLASH), es compatible pin a pin con el chip de comunicación y ofrece un menor consumo.

Las características principales que aporta el nuevo microcontrolador MSP430F2618 son las siguientes:

- Rango de voltaje de alimentación: **1.8V-3.6V**.
- Ultra-bajo consumo:
  - Modo Activo: **280µA**.
  - Modo Standby: **1.6µA**.
  - Modo off: **0.1µA**.
  - Tiempo de conmutación entre Standby y Activo: **<1µs**.
- Memoria Flash: **116KB + 256B**.
- Memoria RAM: **8KB**.
- Convertidor analógico-digital (ADC) de **8** entradas de **12 bits**.
- Hasta **5** modos de bajo consumo.
- **2** Timers de 16 bits, con **3** y **7** registros de captura/comparación.
- Interfaces de comunicación: **4 USCIs** que pueden funcionar principalmente en modo asíncrono (UART: Universal Asynchronous Receiver-Transmitter) o síncrono (SPI). Incluyen adicionalmente el bus de comunicaciones serie I^2C (Inter-Integrated Circuit).

Para el correcto funcionamiento de la electrónica, se han necesitado varios cristales como referencia para los distintos relojes que se emplean. Disponemos de un módulo básico de reloj que emplea un cristal a 32768 Hz, un oscilador interno de baja frecuencia y bajo consumo, un oscilador interno controlado digitalmente (DCO: Digital Controlled Oscillator) y un oscilador de alta frecuencia. El módulo básico de reloj se emplea en sistemas de bajo coste y bajo consumo y proporciona las siguientes señales de reloj: el reloj auxiliar (ACLK: Auxiliary Clock), reloj principal (MCLK: Main Clock) y el reloj sub-principal (SMCLK: Sub-Main Clock).

4.2.2 - Modos de bajo consumo del MSP430

El MSP430 dispone de un modo activo y cinco modos de bajo consumo. Un evento de interrupción puede despertar al microcontrolador de uno de los cinco modos de bajo consumo, atender solicitudes de cambio de estado o retornar a un modo de bajo consumo a partir del programa de atención a la interrupción.

Los siguientes modos de operación[14] pueden ser configurados por software:

- **Modo activo** (AM).
  - Todos los relojes activos.
- **Modo de bajo consumo 0** (LPM0). Entre 87-105 μA.
  - CPU desactivada.
  - ACLK y SMCLK continúan activos.
  - MCLK desactivado.
- **Modo de bajo consumo 1** (LPM1). Entre 40-55 μA.
  - CPU desactivada.
  - ACLK y SMCLK continúan activos. MCLK desactivado.
  - DCO's dc-generator está desactivado si el DCO no se emplea en el modo activo.
- **Modo de bajo consumo 2** (LPM2). Entre 25-36 μA.
  - CPU desactivada.
  - MCLK y SMCLK desactivados.
  - DCO's dc-generator continua activado.
  - ACLK continúa activo.
- **Modo de bajo consumo 3** (LPM3). Entre 0.6-1.2 μA.
  - CPU desactivada.
  - MCLK y SMCLK desactivados.
  - DCO's dc-generator desactivado.
  - ACLK continúa activo.
- **Modo de bajo consumo 4** (LPM4). Entre 0.2-0.5 μA.
  - CPU desactivada.
  - MCLK y SMCLK desactivados.
  - DCO's dc-generator desactivado.
  - ACLK desactivado.
  - Cristal del oscilador detenido.

El empleo adecuado de los diferentes modos de consumo resultará fundamental para conseguir una larga autonomía en nuestros dispositivos. Por tanto, resulta especialmente importante en aquellos sistemas en los que el funcionamiento de los dispositivos se realiza a tiempo parcial y en los que las tareas más importantes se realizan en una porción muy pequeña del tiempo.

Como veremos en el capítulo siguiente, no siempre será posible emplear el modo de consumo que nosotros deseemos; por ejemplo, el modo LPM4 que proporciona el mayor bajo consumo, ya que habitualmente el stack que implementemos (Z-Stack en nuestro caso) decidirá de forma taxativa qué modo de consumo emplear ante las diferentes situaciones que puedan ocurrir.

[14] Los valores de consumo de los diferentes modos de operación son válidos para temperaturas que oscilan entre -40°C y 85°C.

4.2.3 - Chip de comunicaciones CC2520

El CC2520 [15], creado y distribuido por Texas Instruments junto a sus microcontroladores, es el transceptor de RF de segunda generación preparado para implementar ZigBee/IEEE 802.15.4, que opera en la banda de frecuencia libre ISM de 2.4 GHz a una tasa binaria de 250 Kbps. Está diseñado para aplicaciones inalámbricas de bajo consumo. Constituye una excelente opción de bajo costo y altamente integrado para una comunicación inalámbrica robusta. Las principales características del chip de comunicaciones están resumidas en la siguiente tabla:

	Mínimo	Típico	Máximo	Unidades
Rango de frecuencias RF	2394		2507	MHz
Transmisión				
Tasa binaria de transmisión	250		250	Kbps
Tasa de transmisión ensanchada		2		MChip/s
Potencia de transmisión		5		dBm
Recepción				
Sensibilidad		-98		dBm
Regulador de tensión				
Voltaje de entrada	1.8		3.8	V
Tiempo de encendido				
Consumo				
Modo LPM1		175	250	µA
Modo LPM2		30	120	µA
Modo Activo		16	19	mA
Modo de transmisión	25.8(@0dBm)		33.6 (@5dBm)	mA
Modo de recepción		18.5		mA

Tabla 4-1: Características CC2520

En la Tabla 4-1 se pueden observar los distintos modos de funcionamiento del chip desde un punto de vista de consumo energético. Como es de esperar, los mayores consumos tienen lugar en los momentos de transmisión y recepción. Debido a que, como ya se ha comentado anteriormente, la duración de las baterías es un requisito fundamental, buscaremos minimizar la actividad radio del sistema de modo que intentaremos mantener el máximo tiempo posible al chip de comunicaciones en uno de los estados de menor consumo (LPM1 o LPM2). En cuanto al resto de parámetros, son típicos dentro de esta tecnología aunque se ofrecen mejoras tanto en la potencia de transmisión (5 dBm cuando la especificación IEEE 802.15.4 marca -3dBm) como en sensibilidad de recepción (-98 dBm cuando la especificación marca -85 dBm).

Como sucede con la mayoría de chips integrados, el CC2520 requiere de una electrónica externa para su funcionamiento (ver Figura 4-2). Como se puede apreciar, apenas requiere más que un condensador de desacoplo, una resistencia de polarización para generar la corriente adecuada de alimentación, un circuito oscilador y el balun[15] de acoplo a la antena.

Aunque el chip está fabricado para ser directamente compatible con la especificación IEEE 802.15.4, algunos parámetros (potencia de transmisión o formato de trama) pueden ser programados con características diferentes en caso de que fuese necesario.

[15] Balun (**Ba**lanced to **un**balanced) es un dispositivo que se emplea en la adaptación de impedancias.

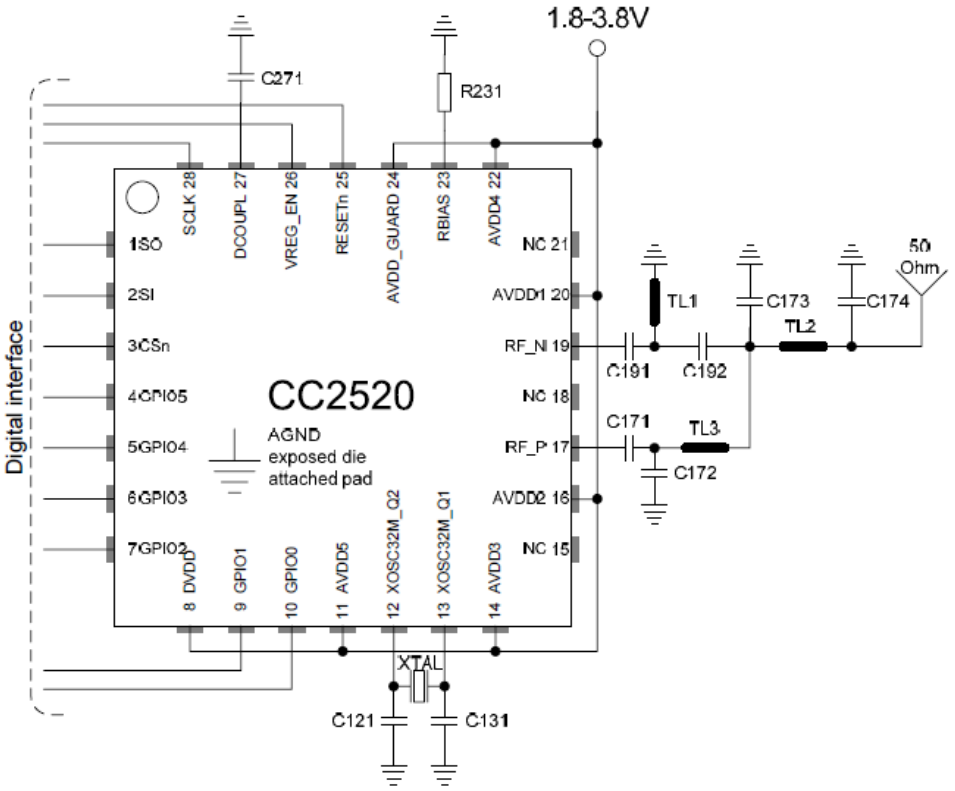

Figura 4-2: Electrónica CC2520

El CC2520 actuará de acuerdo al valor de unos registros que pueden ser cambiados mediante comandos (strobes). Dichos registros se utilizan para establecer la configuración del chip de comunicaciones, para controlar su funcionamiento y para manejar las FIFOs de transmisión/ recepción. Se accede a ellos desde el microcontrolador a través de una interfaz SPI, eligiendo si se hace para escribirlos o leerlos. El CC2520 permite adicionalmente que los comandos puedan ser accionados mediante pines de propósito general de entrada/salida (GPIO: General Purpose Port Input/Output), lo que permite, por ejemplo, el control por temporizadores.

Las FIFOs de transmisión (TXFIFO) y recepción (RXFIFO) son de 128 bytes cada una y constituyen un elemento fundamental en el proceso de comunicación. Cuando queremos transmitir una trama, debemos escribirla previamente en la FIFO de transmisión para posteriormente enviar el STXON strobe (comando de inicio de transmisión). De forma análoga, cuando una trama es recibida correctamente y es almacenada en la FIFO correspondiente, se manda un aviso al microcontrolador a través de las líneas de control de que ha llegado una trama válida y ha de recogerse de la FIFO de recepción.

Además, se ofrecen otras funcionalidades que facilitan la labor de desarrollo, como la medida de potencia recibida (RSSI: Received Signal Strenght Indicator) o el reconocimiento automático de direcciones. Respecto al reconocimiento automático de direcciones, éste se realiza mediante las directrices marcadas por el estándar IEEE 802.15.4 y depende del tipo de direccionamiento y del tipo de trama. Si no se cumplen las condiciones fijas por el estándar, la trama será desechada de la FIFO de recepción.

4.2.4 - Modos de bajo consumo del CC2520

Como podemos observar en la Tabla 4-1, el CC2520 presenta los siguientes modos de consumo:

- En el **Low Power Mode 2** (LPM2) el regulador digital de tensión está desactivado (VREG_EN=0) y no funciona ningún reloj. Todos los módulos analógicos se encuentran en estado power down.
- En el **Low Power Mode 1** (LPM1) el regulador digital de tensión está activado (VREG_EN=1), pero no hay ningún reloj funcionando. Las señales de power down de los módulos analógicos están controladas por la parte digital.
- En el **modo activo** el regulador digital de tensión está activado (VREG_EN=1) y los relojes funcionan. Las señales de power down de los módulos analógicos están controladas por la parte digital.

Cuando un dispositivo ha estado en el LPM2, el contenido de sus registros se pierde. Por tanto, para volver al modo activo, será necesario realizar un reset o el dispositivo estará en un estado desconocido. El reset puede aplicarse poniendo a cero el pin RESET_n o enviando una instrucción de reset (SRES) a través del SPI.

Antes de entrar LPM2, es recomendable hacer un reset al dispositivo. De esta manera, la configuración será siempre la misma cuando la alimentación de la parte digital sea quitada, y así será menos probable que haya problemas con picos de corriente u otros efectos secundarios. Para poder entrar en el modo LPM2 (menor consumo) deberemos previamente pasar por el modo LPM1.

4.3 - Interfaz MSP430-CC2520

El chip CC2520 opera en base a las órdenes de un microcontrolador. Para conseguir comunicar correctamente ambos chips, se emplean una serie de conexiones entre los pines del MSP430 y el CC2520. En la Figura 4-3 se muestra la interfaz que originalmente utilizaba la placa de SAYME.

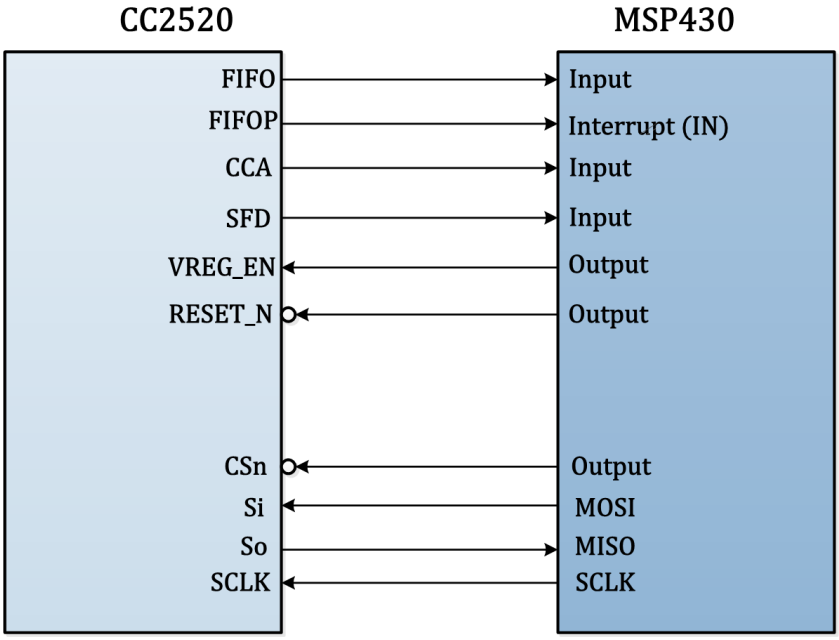

Figura 4-3: Interfaz original MSP430-CC2520

Si bien esta interfaz era adecuada para SWAN, cuando se implementó ZigBee no funcionó según lo esperado, observándose errores durante proceso de asociación de los dispositivos al coordinador.

Una de las tareas más importantes en el desarrollo de este proyecto fue buscar una solución a este problema. Para ello, se analizó el comportamiento del Z-Stack, configurado específicamente para el kit de desarrollo CC2530 de Texas Instruments. Dicho kit emplea el mismo modelo de chip de comunicaciones y de microcontrolador por lo que fue necesario entender su conexionado para poder rediseñar la placa.

Tras un largo proceso de estudio se observó que eran necesarias dos líneas adicionales, una de ellas con capacidad para generar interrupciones, que no habían sido contempladas en el diseño original. Estos cambios obligaron a fabricar un modelo de placa con la nueva interfaz.

Se observó que los problemas tenían su origen en que faltaba asignar la línea con interrupciones TX_ACK_DONE/TX_FRM_DONE. Ésta se emplea para avisar al microcontrolador de que se ha recibido correctamente una trama, por lo que, al no estar contemplada en el diseño original, se producían errores en el coordinador de forma que éste no sabía si un end-device le había mandado una trama de asociación para unirse a la red.

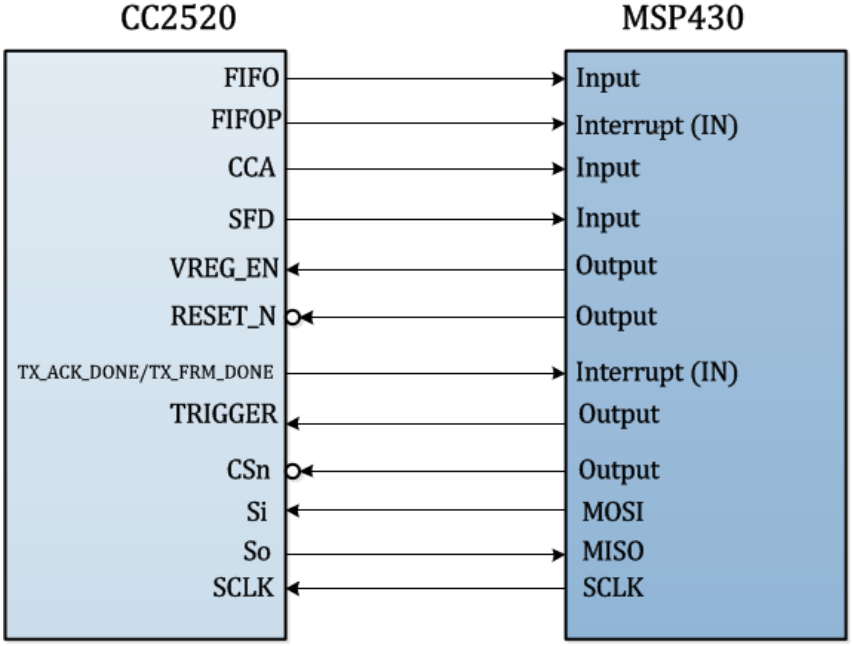

Figura 4-4: Interfaz MSP430-CC2520

A continuación vamos a proceder a explicar el funcionamiento de las líneas mostradas en la Figura 4-4. Se clasifican en tres grupos:

- **Líneas de gestión de eventos**. Empleamos estas líneas para el control del flujo de transmisión y recepción. Cada una de ellas tiene su función definida y nos permite atender correctamente al chip en las diferentes fases de la comunicación.
    - FIFO: Señal que toma el valor uno cuando hay uno o más bytes en la FIFO de recepción (RXFIFO). Permanecerá en dicho valor hasta que ésta se vacíe.
    - FIFOP: Las transiciones de esta señal nos indican o bien la recepción completa de una trama o un exceso de datos no leídos en la RXFIFO mediante un valor umbral (Frame Filtering).

- SFD: Se activa justo después de recibir/enviar completamente el campo de SFD de una trama.

Cuando se recibe una trama, ésta se almacena en la FIFO de recepción (FIFORX). En el momento en el que se detecta el comienzo de una trama (campo SFD), la línea SFD cambia a valor alto. Como se observa en la Figura 4-5, la línea de FIFO toma valor alto cuando se recibe el primer byte de longitud de trama, ya que ni el preámbulo ni el SFD se guardan en memoria. En el caso de que haya varias tramas en la FIFORX, la línea no dejará el valor alto hasta que se hayan leído todas. Por otro lado, la línea FIFOP se utiliza para controlar el volumen de datos pendientes. En el caso de sobrepasar este umbral, la línea FIFOP lo indica poniéndose a valor alto. De igual manera, la línea FIFOP se activa en el caso de que llegue el último byte de una trama nueva y bajará una vez que sea leído un byte de la FIFO de recepción.

Para el caso de la transmisión, la única línea que se utiliza es SFD. Con ella se indica la transmisión completa del preámbulo y el marcador del comienzo de trama.

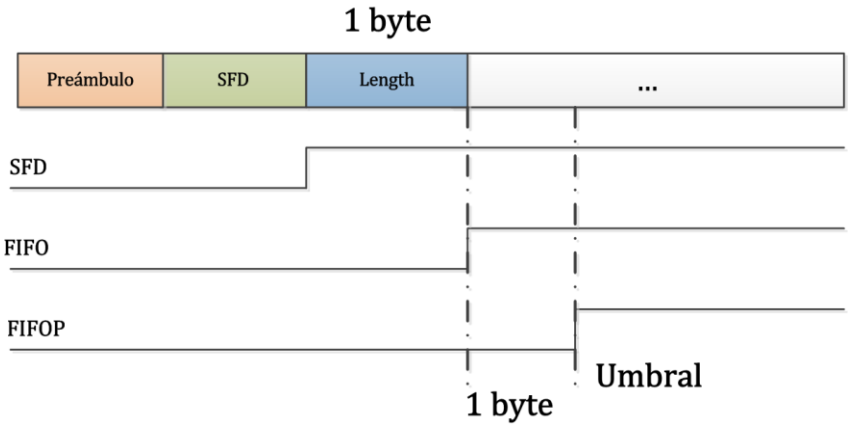

Figura 4-5: Líneas de gestión

- **Interfaz SPI**. Permite a un único dispositivo maestro la gestión de varios esclavos.
  - CSn (Chip Select): Activa el dispositivo esclavo. En nuestro caso, el microcontrolador hace de maestro y el chip de comunicaciones de esclavo.
  - Si (Slave Input): Entrada de datos serie desde el microcontrolador hasta el chip de comunicaciones.
  - So (Slave Output): Salida de datos serie desde el chip de comunicaciones hasta el microcontrolador.
  - SCLK (Serial Clock): Entrada para el reloj de referencia proporcionado por el microcontrolador.
- **Otras conexiones**.
  - VREG_EN: El habilitador del regulador de tensión. Permite utilizar modos de bajo consumo.
  - RESET_N: Esta señal la emplea el microcontrolador para hacer un reset al CC2520.
  - TX_ACK_DONE/TX_FRM_DONE: Se emplea para avisar al microcontrolador de que la transmisión de una trama ha sido correcta, ya sea de información (FRM) o de confirmación (ACK).
  - TRIGGER: Línea definida para versiones futuras. Actualmente no tiene utilidad y puede ser omitida.

4.4 - Placa propietaria de SAYME

En la Figura 4-6 podemos apreciar la placa desarrollada por SAYME empleada en el presente proyecto. Es el primer modelo que incluye la interfaz adecuada entre el MSP430F2618 y el CC2520 para implementar el stack de ZigBee. Mediante la programación software podemos configurarla para que actúe como coordinador, router o end-device sin necesidad de variar ningún componente hardware.

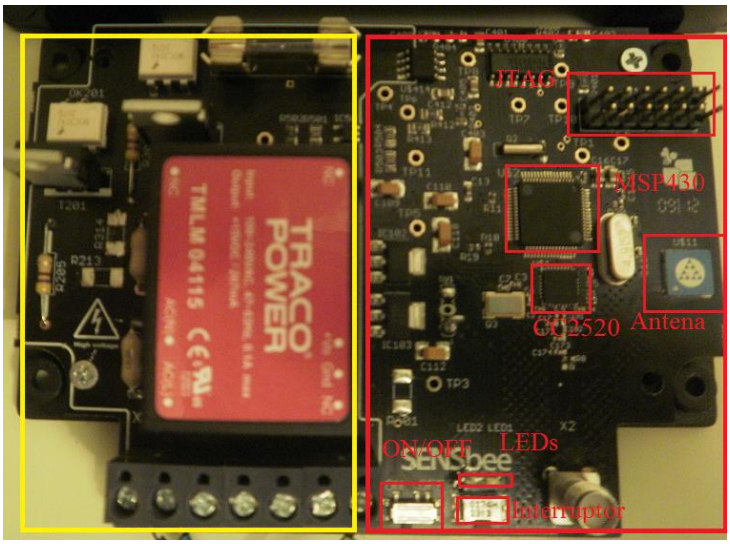

Figura 4-6: Placa de SAYME

Estas placas fueron desarrolladas en su origen para trabajar como coordinadores o routers en una aplicación de iluminación con farolas. Para implementar los end-devices se emplea otro modelo totalmente distinto preparado específicamente para el bajo consumo. Durante el desarrollo del proyecto, se utilizó indistintamente este primer modelo experimental para implementar un coordinador, un router y un end-device.

En la Figura 4-6, el área delimitada en color amarillo se denomina subcircuito de alimentación y su función más importante es transformar los 220V (AC) que le llegan a su entrada a los 3.3V (DC) necesarios para alimentar la parte digital del circuito, representada en color rojo. Adicionalmente se ofrecen otras características para la aplicación de farolas como la opción de *dimmer*[16], pero no han sido empleadas en este proyecto. En la parte digital tenemos los elementos hardware que han sido descritos en este capítulo (microcontrolador y chip de comunicaciones) y otros componentes como:

- 2 LEDs (rojo y amarillo).
- Un pulsador.
- Un interruptor de encendido/apagado.
- Una antena integrada.
- Un conector SMA para emplear una antena externa.
- Un puerto JTAG para programar el microcontrolador.

En el siguiente capítulo explicaremos como deben programarse las librerías HAL (Hardware Abstraction Layer) para poder utilizar todos estos componentes hardware mediante software.

[16] Los dimmer o dímmer permiten regular la energía en una o varias lámparas con el fin de variar la intensidad de luz que emiten.

Capítulo 5: IMPLEMENTACIÓN SOFTWARE

En este capítulo se procederá a introducir las herramientas software con las que se ha elaborado el proyecto. Después, se estudiará el funcionamiento de la implementación de ZigBee (Z-Stack) desarrollada por Texas Instruments. Una parte muy importante en el desarrollo de la integración fue la reprogramación de las librerías HAL proporcionadas por TI en su kit de desarrollo, de modo que fuera posible implementar el protocolo ZigBee en una placa propietaria diseñada por SAYME. Por tanto, se llevará a cabo una descripción de todo el proceso de programación y las dificultades que platea la migración a una tecnología distinta. Por último, plantearemos el desarrollo de una aplicación ZigBee de prueba.

5.1 - Sistema de desarrollo

5.1.1 - IAR Workbench

El IAR Workbench [16] nos ofrece un entorno de programación completo para microprocesadores en el que se incluye el compilador de C, el ensamblador, el depurador y demás elementos necesarios para el desarrollo del software. Además, ofrece distintas versiones adaptadas a las distintas familias de microprocesadores, por lo que la versión empleada en el presente proyecto es la dedicada a la familia MSP430 de Texas Instruments.

Como se observa en la Figura 5-1, el programa presenta una intuitiva interfaz gráfica en la que podemos ver de forma organizada las diferentes carpetas que forman el proyecto. Dichas carpetas contienen los distintos ficheros de aplicación, configuración, compilación y las librerías necesarias para el desarrollo de un proyecto.

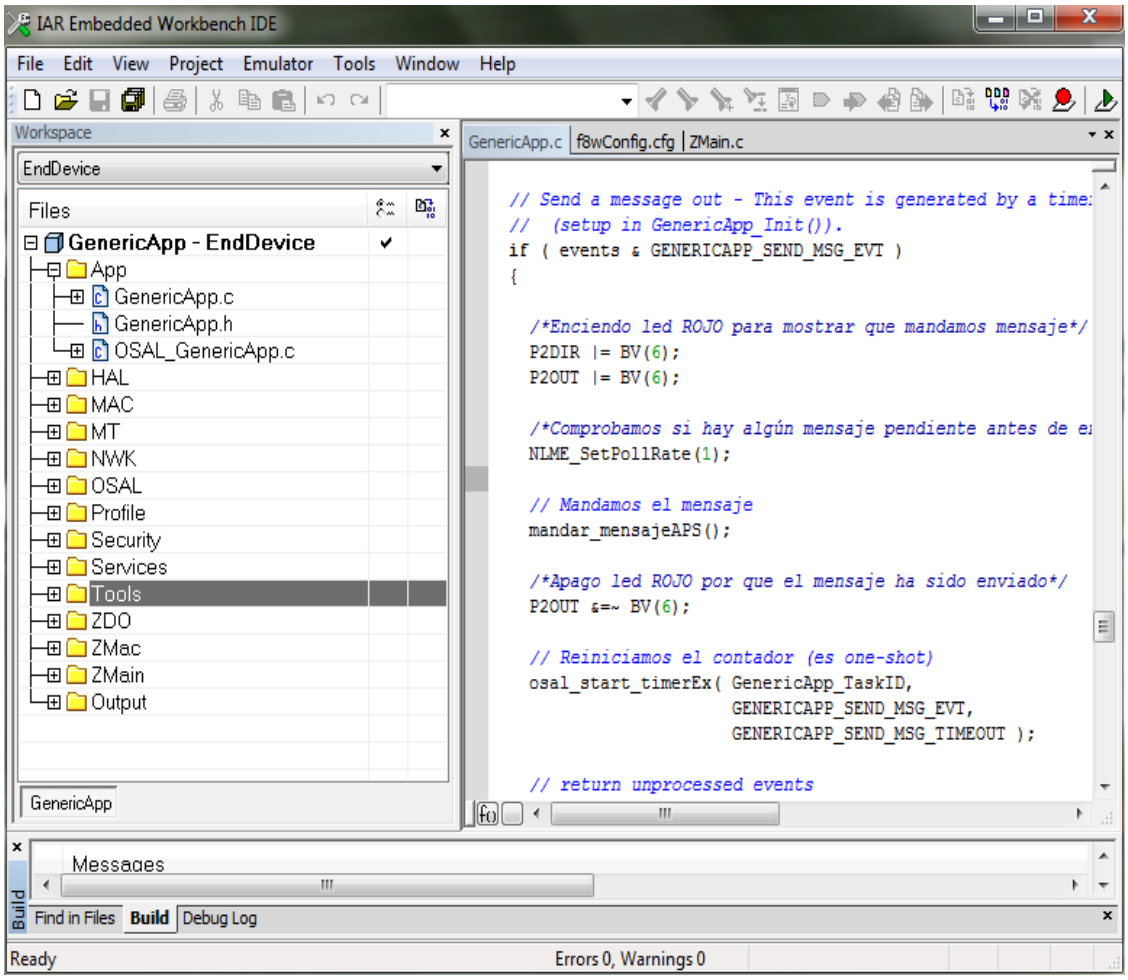

Figura 5-1: IAR Workbench

Como sucede en la mayoría de entornos de programación, IAR Workbench permite la traducción del proyecto en C a un archivo ejecutable por el microcontrolador. Para llevar a cabo esta tarea, el programa proporciona los siguientes elementos: el compilador, el enlazador (linker), el ensamblador y el depurador (debugger). A continuación procederemos a describir que tarea desempeñan cada uno de ellos.

5.1.1.1 - El compilador

El compilador traduce el programa escrito en C a un código objeto que el microcontrolador será capaz de interpretar. Generalmente el código objeto es lenguaje máquina, pero también puede ser un código intermedio (bytecode) o simplemente texto. Este proceso de traducción se conoce como compilación. Como puede observarse en la Figura 5-2, la etapa de compilación puede dividirse en distintos pasos:

- **Scanner**. Una primera etapa de análisis lexicográfico en el que se prueba si las expresiones empleadas en el archivo son correctas desde un punto de vista léxico.
- **Parser**. Una vez comprobado que los símbolos empleados son válidos, el siguiente paso consiste en comprobar si las asociaciones son correctas.
- **Generador de código intermedio**. Antes de generar el código objeto se pasa por una etapa intermedia en la que se crea un archivo con un código con un nivel de abstracción que se encuentra entre el alto nivel y el lenguaje máquina.
- **Optimizador**. A partir del código intermedio se realiza una optimización del código. Por ejemplo, se eliminan todas las variables no usadas y se reordena el código de forma que sea más eficiente. Los criterios empleados en la optimización pueden configurarse.
- **Generador de código objeto**. El último paso es crear el código objeto a partir del código intermedio optimizado.

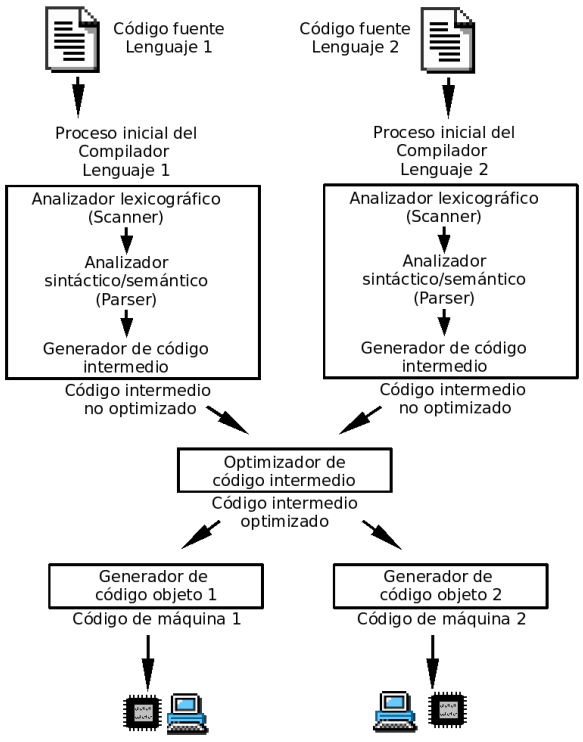

Figura 5-2: Proceso de compilación

El proceso de compilación se realiza archivo a archivo, por tanto, se generará un código objeto por cada archivo en C que contenga nuestro proyecto.

5.1.1.2 - El enlazador

Una vez superada la etapa de compilación dispondremos de varios códigos objeto dentro de nuestro proyecto, uno por cada archivo en C, denominados módulos. En los archivos de código objeto la mayor parte del contenido se encuentra en forma de definición de símbolos. Éstos pueden ser:

- **Definidos o exportados**. Funciones o variables que están presentes en el módulo definido por el código objeto y que deberían ser accesibles para el uso de otros módulos.
- **Indefinidos o importados**. Funciones o variables que son referenciados por el módulo pero que no se encuentran definidas en el código objeto.

La tarea del enlazador (linker) es la de resolver las referencias no resueltas, es decir, aquellas que se corresponden a los símbolos importados. Además, resuelve la asociación de direcciones del código objeto para su transferencia a la memoria del microprocesador. Por último, genera un único archivo ejecutable en el que se recoge la información de todos los archivos, sus referencias cruzadas y las direcciones de memoria para la programación del microcontrolador.

5.1.1.3 - El depurador

El depurador (debugger) es una herramienta que nos permite llevar el control de la ejecución de nuestro programa. Ofrece funciones como correr el programa pasa a paso, parar el programa (breacking), es decir, pausar el programa para examinar el estado actual en cierto evento o instrucción especificada por medio de un punto de ruptura (breakpoint), y el seguimiento de los valores de las variables.

Resulta habitual que en los microcontroladores exista un debugger integrado, lo que se conoce como In-Circuit Debugger. Generalmente se controla mediante ciertas líneas de la interfaz JTAG que nos permiten comunicarnos con el microcontrolador para programarlo, permitiendo también poder aplicar las herramientas para el depurado.

Durante las fases del desarrollo software, el debugger es una herramienta fundamental ya que mediante su empleo es posible detectar rápidamente fallos y solucionarlos. Además, permite comprobar el estado del microcontrolador durante la fase de ejecución, observando en tiempo real el estado de los parámetros. Por último, permite cargar también el ejecutable en la memoria flash de microcontrolador.

5.1.2 - Packet Sniffer

Es un software proporcionado por Texas Instruments que mediante una placa externa permite de capturar tramas IEEE 802.15.4/ZigBee que viajan por el aire, para luego mostrarlas gráficamente desglosadas en campos, lo que ayuda a depurar el código de nuestra aplicación y a comprobar la correcta transmisión/recepción de las tramas.

Para el desarrollo del proyecto, la empresa SAYME colaboró proporcionando la placa de captura (sniffer) CC2531 Dongle [17] con la que es posible analizar los paquetes de la última especificación ZigBee (ZigBee-2007).

Podemos elegir para escuchar de entre los 16 canales disponibles aquél en el que estemos realizando el proceso de comunicación. El sniffer capturará todas las tramas del canal que hayamos seleccionado, independientemente de la procedencia de estas. Las tramas se muestran sucesivamente por pantalla con el instante de tiempo en que tienen lugar (tomando como

referencia el inicio de la captura), un indicador de calidad de la señal recibida (LQI o RSSI) y la comprobación de errores en la trama (FCS). Opcionalmente permite realizar filtrados de las tramas según si el FCS es válido o no, tipo de tramas, etc.

Como se puede apreciar en la Figura 5-3, el Packet Sniffer traduce el payload de las tramas de nivel enlace para que podamos observar la información que contienen las diferentes capas que implementa ZigBee (red, seguridad y aplicación) de forma sencilla y cómoda. Por ello, en todo momento podemos tener una idea clara de qué y quién está trasmitiendo, visualizando si la comunicación en la red se está realizando como nosotros deseamos.

Además, genera un cronograma en tiempo real con el que es bastante sencillo seguir la secuencia de envío de tramas y comprobar posibles fallos en la misma.

Figura 5-3: Packet Sniffer

5.1.3 - Pila de protocolo: Z-Stack

Z-Stack [18] es la implementación desarrollada por Texas Instruments de la especificación ZigBee-2007. Está certificado como una ZigBee Compliant Platform (ZCP) por la ZigBee Alliance y permite emplear cualquiera de las dos especificaciones definidas en ZigBee-2007: ZigBee y ZigBee PRO. Por su parte, Z-Stack ha sido específicamente diseñado para la familia de chips CC2530 (SoC) y para el extenso abanico de microcontroladores MSP430 en combinación con los chips de comunicaciones CC2520.

Como veíamos anteriormente en la Figura 5-1, Texas Instruments proporciona el módulo de Z-Stack en formato de proyecto IAR. Como hemos comentado, este proyecto se divide en carpetas que simulan las capas del stack completo. A continuación, se detallan las características y funcionalidades de las carpetas más importantes para entender el Z-Stack y ser capaces de modificarlo si fuera necesario.

- **App**. En esta carpeta se define el funcionamiento de nuestra aplicación y sus librerías.
- **HAL**. Carpeta que incluye las librerías HAL (Hardware Abstraction Layer). Como se verá más adelante, estas librerías simplifican la programación software, ya que aíslan el hardware del programador.
- **MAC**. Aquí están los ficheros que definen el nivel de enlace del IEEE 802.15.4.
- **NWK**. Se incluyen los ficheros que definen el nivel de red (NWK) de ZigBee.
- **OSAL**. Esta es una de las carpetas más importantes. Implementa un sistema operativo reducido que se encarga de organizar y atender los eventos que se producen durante la ejecución del programa. Gracias a su empleo se puede programar el software independientemente del sistema que haya por debajo.

- **Profile**. Dentro de esta carpeta se encuentran los ficheros que definen el AF (Application Framework). Se incluyen las funciones para el registro de endpoints, envío de paquetes de aplicación, etc.
- **Tools**. Aquí se encuentran los ficheros de configuración (.cfg) que especifican los parámetros de los dispositivos de la red: coordinador, routers y end-devices. También hay un fichero de configuración en el que se especifican los parámetros para la formación de la red y su gestión. Por ejemplo, los canales que se van a escanear, tiempo entre retransmisiones, intervalos de polling, etc.
- **ZDO**. Dentro de esta carpeta se encuentran los ficheros que definen la interfaz de la aplicación del dispositivo ZigBee y los atributos de configuración.
- **Zmain**. En esta carpeta se incluyen los ficheros de inicialización hardware de la placa. Además, tenemos el fichero main (ZMain.c) de nuestro programa. Él es el encargado de arrancar la aplicación y de pasarle el control a la capa OSAL, quien se encargará de gestionar el resto de la ejecución.

Una vez descritas las carpetas de un proyecto, vamos a proceder a describir las principales herramientas que proporciona el Z-Stack para el desarrollo de aplicaciones. En concreto: HAL (Hardware Abstraction Layer), OSAL (Operating System Abstraction Layer), rutinas de interrupción (ISR) y la gestión de la alimentación (Power Saving).

5.1.3.1 - OSAL

El OSAL (Operation System Abstraction Layer) proporciona una interfaz de programación de aplicaciones (API: Application Programming Interface) que permite que un determinado software pueda funcionar independientemente del sistema operativo, kernel o entorno de tareas (incluyendo ciclos de control y sistemas de interrupción) empleado. Para ello, ofrece las siguientes funcionalidades:

- Registro, inicialización y puesta en marcha de tareas.
- Intercambio de mensajes entre tareas.
- Sincronización de tareas.
- Gestor de interrupciones.
- Temporizadores (timers).
- Gestión de memoria.

Debido a las limitadas funcionalidades, el OSAL no es, estrictamente hablando, un sistema operativo.

A continuación se detallarán algunas de las funciones del OSAL empleadas en el desarrollo software del presente proyecto. No obstante, es necesario explicar previamente una serie de conceptos que son empleados por el OSAL:

- **Tarea**. Fragmento de código (thread) que desarrolla una determinada función. Toda tarea debe contar con una sección de inicialización y otra de ejecución. Además, una vez comenzada la ejecución la tarea no terminará hasta que se haya completado.
- **Evento**. Acción que debe ser completada por una tarea. Los eventos causan que se inicie la sección de ejecución de una tarea. Es habitual que los eventos sean temporizadores o triggers que se disparan ante determinadas situaciones (por ejemplo, la recepción de una trama).
- **Mensaje**. Sirven para intercambiar información entre tareas.

De modo que, como se observa en la Figura 5-4 , las tareas se inicializan por eventos y se comunican entre sí mediante el intercambio de mensajes.

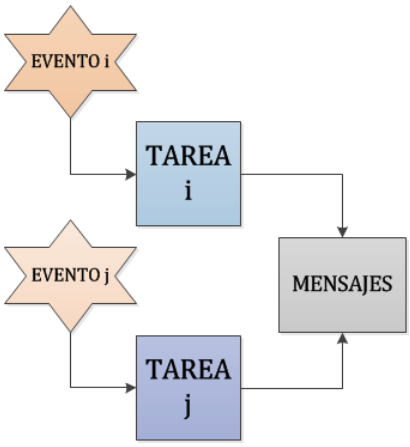

Figura 5-4: Eventos, tareas y mensajes

Por su parte, las funcionalidades del OSAL empleadas en el proyecto fueron:

- **Gestión de mensajes.** La API de gestión de mensajes proporciona un mecanismo para el intercambio de mensajes entre tareas o procesos de distintos entornos. Por ejemplo, una rutina de atención a interrupciones o funciones que son llamadas dentro de un ciclo de control. Estas funciones de la API permiten trabajar con los buffers de memoria, mandar mensajes de comandos a otra tarea y recibir los mensajes de respuesta.
- **Sincronización de tareas.** Esta API permite a una tarea esperar a que un evento tenga lugar y devolverle el control mientras espera. Estas funciones pueden ser empleadas para establecer eventos de una determinada tarea y sus correspondientes notificaciones.
- **Gestión de temporizadores**. Mediante esta API podemos establecer temporizadores (timers) tanto para tareas internas (Z-Stack) como para tareas externas (nivel aplicación). Debido a las limitaciones hardware, no podremos emplear timers con incrementos menores a un milisegundo. La aplicación ZigBee propuesta en el presente proyecto hace uso de temporizadores para controlar las transmisiones de información y para gestionar el mecanismo de encuesta que emplea el end-device, como se ha explicado en la sección dedicada al funcionamiento del protocolo ZigBee. Así, mediante el uso de la función *osal_start_timerEx* podemos inicializar un timer, por ejemplo, para transmitir periódicamente una trama de información.
- **Gestión de interrupciones**. Las funciones de esta API permiten a una tarea asociar una rutina de servicio a una determinada interrupción. Dentro de una determinada rutina de servicio, pueden existir eventos que generen otras nuevas tareas. Con la función *osal_int_enable* habilitamos una interrupción. Una vez establecida, cuando se produzca la interrupción se llamará a la rutina de atención asociada a dicha interrupción.
- **Gestión de tareas.** Esta API se emplea para añadir y gestionar tareas en el OSAL. Cada tarea se compone de una función de inicialización y de una función de procesamiento de eventos. El sistema emplea la función *osal_init_system* para inicializar el OSAL. Cuando vaya a inicializar una tarea empleará la función *osalInitTask* y después llamará a un procesador de eventos para cada tarea donde tratará cada evento de forma distinta.
- **Gestión de memoria.** Esta API representa un sistema de asignación de memoria. El sistema utiliza la función *osal_mem_alloc* para gestionar la memoria.
- **Gestión de energía.** El sistema proporciona a las aplicaciones/tareas una forma de notificar al OSAL cuando es seguro desactivar el chip de comunicaciones y determinadas partes del hardware del MCU para entrar, cuando sea necesario, en un modo de bajo consumo. Mediante la función *osal_pwmgr_state* podemos cambiar o establecer el modo de ahorro energético en un dispositivo. Por otro lado, también se emplea la función *osal_pwrmgr_task_state* para cambiar el estado energético asociado a una determinada.
- **Memoria no volátil.** El sistema proporciona una forma de almacenar la información de las aplicaciones dentro de la memoria persistente del dispositivo. Además, puede ser empleada por el stack para almacenar determinados datos requeridos por la especificación

de ZigBee. Para leer y escribir en la memoria, el OSAL proporciona las funciones *osal_nv_write* y *osal_nv_read* respectivamente, sin tener en cuenta el tipo de memoria (EEPROM o flash).

5.1.3.2 - HAL

La capa de abstracción de hardware (HAL: Hardware Abstraction Layer) es un elemento del sistema que funciona como una interfaz entre el software y el hardware del sistema, proveyendo una plataforma de hardware consistente sobre la cual correr las aplicaciones. De esta forma, aíslan al desarrollador del tratamiento hardware. Abarca numerosos aspectos, desde el control de los módulos USCI hasta las macros que implementan, por ejemplo, la comunicación SPI. Habitualmente son proporcionados por el fabricante, aunque generalmente los ofrecen los proveedores de microcontroladores para el manejo de sus chips.

La versión de Z-Stack que proporciona Texas Instruments presenta unas librerías HAL preparadas para el empleo del kit de desarrollo CC2520DK. Permiten gestionar mediante funciones los siguientes componentes: el conversor analógico-digital (ADC: Analog-Digital Converter), una pantalla (LCD: Liquid Crystal Display), los LEDS, los botones (KEYs), modo dormido (SLEEP), los temporizadores (TIMERs), los puertos serie (UARTs: Universal Asynchronous Receiver-Transmitter), y la amplificación (LNA: Low-Noise Amplifier). Además, se encarga de mapear las líneas de control que emplea el microcontrolador para gestionar el chip de comunicaciones. Por otro lado, también gestiona otros elementos del microcontrolador como el reloj, los vectores de interrupción o el DCO.

El CC2520DK utiliza el mismo hardware (microcontrolador y chip de comunicaciones) que la placa de la empresa SAYME, pero incluye adicionalmente otros componentes como una pantalla LCD o un Joystick de los que no dispone la placa.

Por tanto, el primer paso para desarrollar una aplicación ZigBee fue adaptar las librerías HAL originales del Z-Stack a la placa proporcionada por SAYME. Para ello, se modificó el fichero **hal_board_cfg.h** que contiene la configuración de los componentes hardware de la placa. De modo que se procedió a eliminar aquellos elementos no disponibles en nuestra placa (LCD, Joystick, botones, etc.) y se mapearon aquellos que si teníamos (LEDs, pulsador, etc.). No obstante, otros componentes no necesitaron ningún cambio (UART y ADC) ya que estaban configurados de igual manera que en el kit de desarrollo. También, se comprobó que las macros para la gestión de la señal de reloj (cristal o DCO) eran adecuadas a las especificaciones de nuestra placa.

En el siguiente paso, se modificó el archivo **hal_mac_cfg.h** para mapear vía software las líneas de control que, como explicamos en el capítulo anterior, utiliza el microcontrolador para controlar el chip de comunicaciones. Merece especial atención comentar que, en la realización de este proceso, se detectó que eran necesarias líneas de control adicionales que no estaban contempladas en el diseño de la placa de SAYME. Por tanto, fue necesario modificar el hardware de placa para añadir las nuevas líneas de control (TX_ACK_DONE/TX_FRM_DONE y TRIGGER) y así poder implementar el Z-Stack. Esta tarea provocó que se tuvieran que fabricar nuevos modelos de la placa con el consecuente tiempo de espera. No obstante, no podemos olvidar que aunque SWAN emplee los mismos componentes hardware que ZigBee, la interfaz para gestionarlo es diferente y, por tanto, es necesario redefinirla como se explicó en la sección 4.3.

5.1.3.3 - Rutinas de atención a interrupciones (ISR)

Mediante el uso de interrupciones podemos controlar por completo determinados segmentos del programa que son críticos, de forma que es posible atender un determinado evento sin permitir que otro de menor prioridad pueda alterar su funcionamiento. Cuando se produce una

interrupción, ya sea por hardware (pulsar una interruptor, recibir una trama, etc.) o software (un temporizador), el microcontrolador atiende la interrupción mediante una rutina de atención (ISR). Se emplean, por ejemplo, para atender los procesos de transmisión y recepción inalámbricos.

5.1.3.4 - Gestión de la energía (Power Saving)

La gestión eficiente de la energía, como ya se ha comentado en numerosas ocasiones, es un factor clave en todos aquellos módulos que emplean baterías. Por esta razón, Texas Instruments proporciona a los desarrolladores un documento [19] sobre cómo gestionar de manera eficiente la energía en la pila Z-Stack cuando empleamos un MSP430 en combinación con el CC2520. Como se explicó en la sección anterior, el microcontrolador MSP430 dispone de cuatro modos diferentes de bajo consumo donde determinadas funciones (periféricos, señales de reloj, temporizadores, etc.) son desactivadas para aumentar la autonomía del dispositivo. Sin embargo, Z-Stack sólo emplea dos de estos modos: LPM3 y LPM4. Para ello, se definen dos estrategias que hacen uso de estos modos: TIMER sleep y DEEP sleep.

El modo TIMER ("dormir por temporización") se utiliza en sistemas que necesitan despertarse periódicamente para realizar una determinada actividad, por ejemplo medir la temperatura, de forma programada. Se busca minimizar el consumo de energía entre los breves períodos de tiempo que hay entre las tareas programas. En el modo TIMER, el microcontrolador entrará en el modo LPM3 cuando no se encuentre activo.

El modo DEEP ("sueño profundo") se emplea cuando no existen actividades programadas. Por esta razón será necesario algún tipo de estimulo externo, por ejemplo pulsar un botón, para que el dispositivo vuelva a estar activo durante el instante de tiempo en el que realizará su función, para luego volver a dormirse. Por tanto, mediante este modo los dispositivos podrán minimizar su consumo durante los largos periodos de inactividad. En este modo, el microcontrolador podrá entrar en el modo LPM4.

Generalmente con el modo TIMER es posible reducir el consumo energético a unos pocos miliamperios en los periodos de inactividad, mientras que con el modo DEEP podemos llegar a consumos del orden de los microamperios. Cuando se emplea el modo TIMER, el OSAL no permite desactivar otros elementos del dispositivo (chip de comunicaciones), por tanto, aunque el microcontrolador emplee el modo LPM3 (consumiendo microamperios), el consumo total se ve incrementado hasta el orden de los miliamperios.

Como se analizó en la sección referente al protocolo ZigBee, sólo los end-devices pueden implementar técnicas de gestión energética puesto que no tienen capacidad de encaminar mensajes como los routers o el coordinador. Estos últimos necesitan siempre estar activos para encaminar mensajes a otros dispositivos, teniendo que mantener además actualizadas sus correspondientes tablas de rutas y de vecinos.

El encargado de gestionar la actividad del sistema es el OSAL en su lazo de control principal. Si una tarea tiene un evento programado y la opción de gestión de energía está activada, entonces el OSAL decidirá si se puede o no dormir. Para que un end-device pueda entrar en un modo de bajo consumo deberán cumplirse las siguientes condiciones:

- Debe activarse la opción de compilación **POWER_SAVING**.
- El descriptor del nodo (ZDO) debe indicar la opción "desactivar RX cuando esté inactivo".
- Todas las tareas gestionadas en el Z-Stack deben permitir ahorro energético.
- No pueden existir tareas en el Z-Stack criticas.
- La MAC no puede tener tareas programadas.

Para reducir el consumo de energía a niveles mínimos, el end-device debe desactivar la mayor parte de su electrónica antes de entrar en un modo bajo consumo. Esto incluye periféricos, transmisor/receptor RF y varios módulos del microcontrolador. Para evitar la pérdida de mensajes mientras está dormido, el end-device necesita que su padre (router o coordinador con el que está

asociado) guarde los mensajes destinados a él, hasta que al despertar, pregunte por ellos. El padre del dispositivo sabe que el end-device preguntará por sus datos debido a que cuando se produjo el proceso de asociación, la trama de association request enviada por el end-device tenía la característica CAPINFO_RCVR_ON_IDLE desactivada.

La decisión de intentar entrar en un modo de bajo consumo se realiza al final del lazo principal del OSAL mediante la función *osal_pwrmgr_powerconserve*, a la que sólo se puede acceder cuando la variable **POWER_SAVING** ha sido activada en el proceso de pre-compilación. Una vez llamada esta función, se comprueba si el dispositivo está alimentado por baterías observando el valor de la variable *pwrmgr_device*. Después, se comprueba la variable *pwrmgr_task_state* para ver que no existe ninguna tarea crítica que impida entrar en un modo de bajo consumo.

Una vez comprobadas estas dos variables, podremos elegir entre el modo TIMER o DEEP dependiendo de qué tipo de aplicación vayamos a desarrollar. El modo DEEP sólo se podrá emplear cuando no exista ningún proceso programado por temporizadores.

Como se ha comentado durante esta sección, la gestión de energía es un proceso controlado exclusivamente por el sistema OSAL con sus respectivas ventajas e inconvenientes. Para ello, el OSAL decide cuando un dispositivo puede "dormir" para ahorrar baterías independientemente de la aplicación. Sin embargo, esta automatización del control energético puede tener efectos adversos cuando se plantea desarrollar aplicaciones de ultra-bajo consumo que requieren de un exhaustivo control de los modos del microcontrolador, ya que el OSAL no permite al desarrollador ningún tipo de gestión.

5.2 - Consideraciones de diseño

En esta sección se explicará la filosofía seguida para el desarrollo de una aplicación sencilla que permita comprobar el funcionamiento del software descrito anteriormente. Por tanto, el objetivo del proyecto fin de carrera no es analizar en profundidad todas las características que nos ofrece ZigBee, sino demostrar su correcto funcionamiento en unas placas propietarias que emplean el mismo microcontrolador y chip de comunicaciones que las placas oficiales de Texas Instruments, comercializadas específicamente para utilizar ZigBee.

Se pretende, por tanto, desarrollar dos aplicaciones que muestren algunas de las ventajas más importantes de ZigBee: autoconfiguración, comunicación multisalto y el descubrimiento de rutas.

5.2.1 - Estrategias de confirmación

Para mostrar un ejemplo de aplicación utilizaremos una red formada por un coordinador, un router y un end-device.

El end-device enviará de forma periódica tramas de información, por ejemplo las medidas de un determinado sensor, al coordinador de la red. Una vez recibidos los datos, el coordinador responderá confirmando la correcta recepción de los datos.

Para ello, se propone un primer escenario en el que el end-device dispone de la suficiente cobertura para enviarle directamente los datos al coordinador. En un segundo escenario, el end-device ya no estará en el área de cobertura del coordinador, por tanto, deberá hacer uso de un router intermedio para que la información pueda llegar hasta él.

Para confirmar que la información ha llegado correctamente a su destino, ZigBee permite emplear confirmación a dos niveles diferentes: enlace y aplicación. La confirmación a nivel enlace

es obligatoria puesto que así lo define el estándar IEEE 802.15.4 y tiene lugar siempre que un nodo recibe un mensaje (sea o no el destino final), mientras que la confirmación a nivel aplicación es opcional, salvo en determinados profiles de carácter comercial como Home Automation. Asegurar que la información ha sido recibida a nivel aplicación resulta lógico: imaginemos una situación en la que un mensaje tiene que atravesar múltiples nodos para llegar a su destino y queremos que el nodo emisor, en nuestro caso siempre un end-device, tenga constancia de que, efectivamente, su mensaje ha sido recibido correctamente por el coordinador. Por tanto, una primera estrategia que se plantea es utilizar ACKs exclusivamente a nivel enlace, como indica el IEEE 802.15.4. En la Figura 5-5 se puede observar cómo funciona este proceso. Cada vez que una trama de datos llega a un nodo, éste envía una trama ACK de nivel enlace al nodo emisor para confirmar localmente la correcta recepción de los datos.

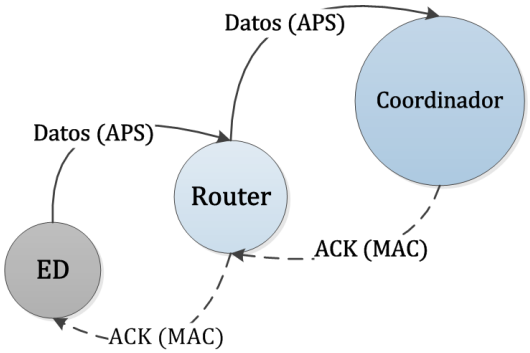

Figura 5-5: Estrategia 1, sólo confirmación a nivel enlace

Como se explicó en la sección referente al IEEE 802.15.4, en caso de que una trama no se reciba en uno de los saltos o no se envíe la correspondiente confirmación, se producirá una retransmisión de los datos. En el caso en el que se supere un número determinado de reintentos fallidos o sin conseguir respuesta, el dispositivo abandonará la red y volverá, después de un determinado tiempo, a intentar unirse a otro dispositivo o a otra nueva red.

La segunda estrategia consistiría en emplear confirmación a nivel enlace y aplicación. Al igual que en la otra opción, la confirmación a nivel enlace nos permitirá saber que nuestro mensaje ha sido recibido correctamente en el siguiente salto, es decir, localmente. En cambio, la confirmación a nivel aplicación indica que nuestro mensaje ha sido recibido correctamente en su destino final. En la Figura 5-6 se aprecia como el coordinador envía una confirmación de nivel aplicación al end-device cuando recibe correctamente los datos de aplicación que éste le ha enviado. Este mensaje de confirmación viajará por la red mesh, en nuestro caso a través de un sólo router, pudiendo volver a utilizar la misma ruta que siguió el mensaje de datos u otra nueva (si la hubiera).

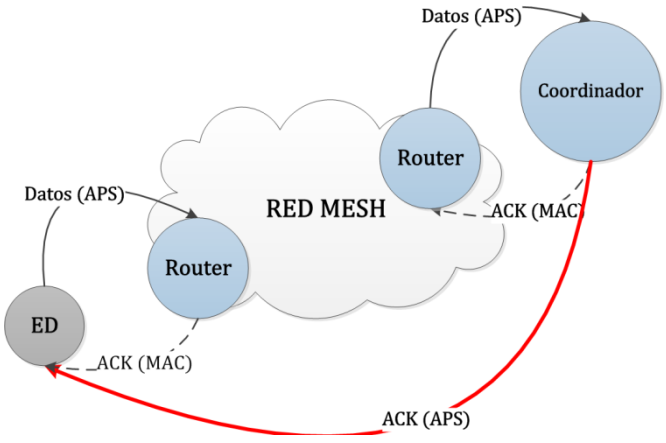

Figura 5-6: Estrategia 2, confirmación a nivel enlace y aplicación

No obstante, emplear dos niveles de confirmación obligará a que sea necesario un mayor intercambio de mensajes entre los diferentes nodos y esto, como resulta evidente, ocasionará un mayor consumo energético en los dispositivos, especialmente en los end-devices, ya que deberán estar más tiempo activos esperando la confirmación.

Una vez analizadas las dos estrategias que podemos llevar a cabo con las confirmaciones, se va a proceder a describir el proceso de formación de la red y asociación de dispositivos en el Z-Stack. Después, se expondrán los intercambios de tramas característicos de los dos escenarios (comunicación directa y red mesh) propuestos, empleando los distintos modos de confirmación (enlace y aplicación) que hemos analizado en esta sección.

5.2.2 - Formación de la red

El coordinador de la PAN es siempre el encargado de inicializar la red en ZigBee. Para el desarrollo del presente proyecto, la red que vamos a implementar utilizará el modo *beaconless* (non-beacon network enabled), como es habitual en ZigBee. Para formar la red el coordinador llevará a cabo la siguiente secuencia:

- **Reinicia su capa de MAC** para volver a la configuración de arranque.
- A continuación realiza una **detección de energía** (ED: Energy Detect) para determinar cuál es el mejor canal, es decir, aquel con menor ruido de los 16 canales disponibles. Se estima que el coordinador puede tardar aproximadamente 8 segundos en escanear todos los canales.
- Después de escoger el mejor canal para formar una red, el coordinador envía tramas de petición (beacon requests) para descubrir si existen otras redes vecinas operando en el mismo canal. Mediante el **escaneado activo** (active scan) el coordinador se asegura de no utilizar el mismo identificador de PAN que el de otras redes vecinas. El escaneado activo puede necesitar varios segundos ya que debe esperar a posibles respuestas (beacon responses) de otros nodos.
- Una vez escogido el mejor canal y comprobado que no existen redes vecinas con el mismo identificador, el coordinador inicializa la red configurando sus **atributos de nivel enlace**: canal, PAN ID y su dirección de red (por defecto 0x0000).

5.2.3 - Proceso de asociación

El proceso de asociación, como se puede observar en la Figura 5-7, es igual para todos los dispositivos que deseen unirse a una determinada red. Durante el primer paso, el dispositivo realizará un escaneado activo (active scan) de todos los canales disponibles mediante el envío de tramas de petición (beacon requests). Cuando el coordinador de una red reciba una de estas peticiones, responderá a dicho dispositivo con una trama con información de su red: el identificador de la PAN, modo de operación de la red, su dirección de red, su dirección de enlace, la versión del stack, etc.

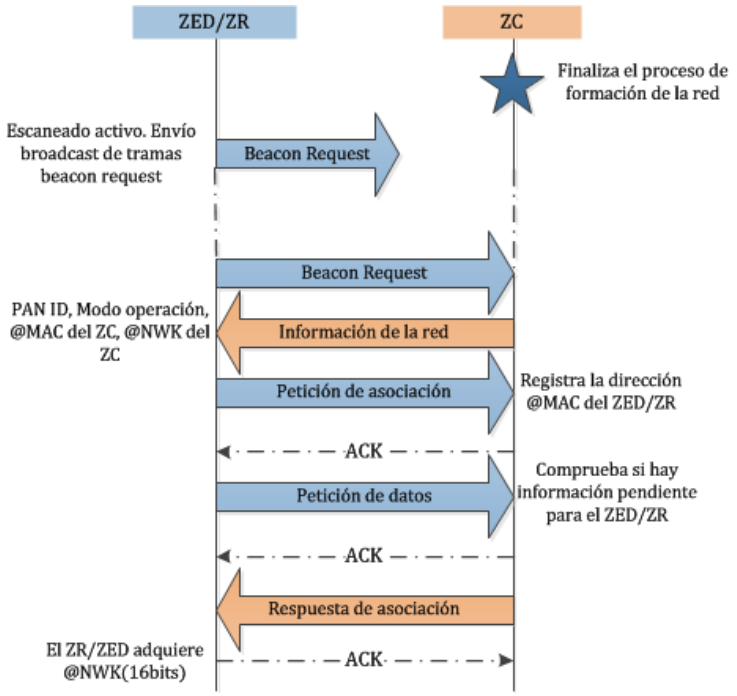

Figura 5-7: Búsqueda y asociación a una red

El dispositivo procesará esa trama y si es posible, intentará unirse a la red. Para ello, enviará una trama de asociación (associate request) al coordinador, éste guardará la dirección del dispositivo y confirmará a nivel enlace la correcta recepción de la petición; a continuación el dispositivo preguntará al coordinador si existe alguna información pendiente para él; una vez recibida la petición y comprobado que no hay mensajes pendientes, el coordinador enviará la confirmación de la asociación (associate response) al dispositivo; por último, el dispositivo confirmará a nivel enlace que ha recibido la trama. Finalizado este proceso, el dispositivo formará parte de la red. Además, periódicamente enviará tramas broadcast para informar de su presencia a los demás nodos.

5.2.4 - Envío de datos a nivel aplicación

Para enviar datos de nivel aplicación a un determinado dispositivo se emplea la función *AF_DataRequest.* Si utilizamos confirmación a nivel aplicación es necesario incluir el parámetro AF_ACK_REQUEST en dicha función para que el coordinador sepa que debe enviar una confirmación de nivel aplicación al end-device que le envió los datos. Por otra parte, el parámetro de dirección destino de esta función es del tipo *afAddrType_t*.

```
typedef struct
{
  union
  {
    uint16      shortAddr;
    ZLongAddr_t extAddr;
  } addr;
  afAddrMode_t addrMode;
  uint8 endPoint;
  uint16 panId;
} afAddrType_t;
```

Figura 5-8: Tipo afAddrType_t

Como se observa Figura 5-8, para enviar un dato a un determinado dispositivo necesitamos saber sus direcciones de endpoint y PAN ID. En la Figura 5-9 podemos observar los diferentes tipos de direccionamiento que, como comentamos en la sección correspondiente a ZigBee, se pueden utilizar: directo (16 bits ó 64 bits), indirecto, de grupo o broadcast.

```
typedef enum
{
  afAddrNotPresent = AddrNotPresent,
  afAddr16Bit      = Addr16Bit,
  afAddr64Bit      = Addr64Bit,
  afAddrGroup      = AddrGroup,
  afAddrBroadcast  = AddrBroadcast
} afAddrMode_t;
```

Figura 5-9: Tipo afAddrMode_t

Para el desarrollo del proyecto se empleó la mensajería directa (con direcciones de 16 bits), es decir, aquella en la que comunicación se produce de dispositivo a dispositivo directamente. La mensajería directa resulta especialmente adecuada cuando se emplean mensajes del tipo unicast. Sin embargo, necesitaremos saber previamente la dirección de red de nuestro destino, en nuestro caso el coordinador (por defecto 0x0000). Por otra parte, como se comentó anteriormente, cualquier envío de datos que se realice a un dispositivo, sea o no el destino final, requerirá siempre de una confirmación de nivel enlace.

5.2.5 - Gestión de tareas

La gestión de las tareas se realiza a través del tratamiento de eventos. Nuestra aplicación define los posibles eventos que queremos tratar y decide en qué momentos se van a producir.

La aplicación propuesta gestiona al menos los siguientes eventos:

- ZDO_STATE_CHANGE. Este evento se produce cuando un dispositivo adquiere durante su inicialización la funcionalidad de coordinador, router o end-device. En el caso de que el dispositivo se inicialice como un end-device, se activará un temporizador para saber cuándo debe transmitir la información. Cuando dicho temporizador llegue al valor estipulado, se generará el evento GENERIC_SEND_MSG_EVT.
- AF_DATA_CONFIRM. Evento encargado de la confirmación a nivel aplicación.
- AF_INCOMING_CMD. Gestiona cómo debe actuar el dispositivo cuando se recibe un mensaje. En nuestro caso, el coordinador enciende uno de los LEDs de la placa.
- GENERICAPP_SEND_MSG_EVT. Evento que se encarga del envío de los datos. Está producido por un temporizador que se inicializa cuando un dispositivo adquiere la funcionalidad de end-device. Una vez enviados los datos, se reiniciará de nuevo el temporizador (one-shot) para la siguiente transmisión.

5.2.6 - Mecanismo de encuesta (polling)

En las redes beaconless los end-devices emplean polling para saber si tienen algún mensaje pendiente de recibir. En este modo, los dispositivos nunca pueden recibir información de nivel aplicación de otro nodo sin antes preguntar por ello. Esta información que esperan recibir puede ser: datos, confirmaciones de nivel aplicación, cambios en la red, comandos de configuración, etc.

Para ello, Z-Stack ofrece al desarrollador dos estrategias para implementar en los end-devices la técnica de polling:

- Manualmente. Nosotros decidimos el momento exacto en que preguntar por los datos, por ejemplo antes de enviar el siguiente.
- Por temporización. En el fichero **f8wConfig.cfg** de configuración del dispositivo, podemos establecer que el end-device envíe periódicamente peticiones datos al dispositivo padre (router o coordinador) para preguntar por mensajes pendientes. Mediante el uso de esta estrategia podemos establecer el intervalo aproximado de tiempo en el que se produce el polling.

En nuestra aplicación hemos utilizado el método manual, ya que de esta forma podemos establecer con mayor precisión los momentos en los que queremos que el end-device pregunte por los datos pendientes.

A continuación, vamos a analizar dos casos hipotéticos que podrían plantearse en nuestra aplicación. Para ello, estudiaremos cómo serán los respectivos intercambios de tramas y qué influencia tiene utilizar confirmación a distintos niveles.

En el primer caso, se comprobará una situación en la que el end-device puede comunicarse directamente con el coordinador. En el segundo caso, el end-device no tendrá suficiente cobertura como para establecer una comunicación con el coordinador y deberá, por tanto, hacer uso de un router intermedio. Mediante estos dos ejemplos mostraremos algunas de las características más importantes del protocolo de comunicación ZigBee.

5.2.7 - Comunicación directa

En la Figura 5-10a podemos observar el caso más sencillo, el end-device envía datos de aplicación, por ejemplo la medida de un sensor, al coordinador de su red de forma periódica. Durante el proceso de asociación a la red, el coordinador informa al end-device de cuáles son sus direcciones de red y endpoint para que éste puede enviarle los datos. En el momento en el que end-device forma parte de la red, éste inicializa un temporizador para transmitir periódicamente la información y lo reiniciará cada vez que envía un dato. El coordinador confirmará cada mensaje recibido correctamente con un ACK de nivel enlace.

Por otra parte, en la Figura 5-10b se representa el caso en el que empleamos adicionalmente confirmación a nivel aplicación. Por tanto, para que el end-device pueda recibir el ACK de aplicación será necesario que éste realice previamente un polling al coordinador. Con el fin de gestionar de manera eficiente la red y el consumo energético de los end-devices, es muy importante reducir al mínimo el número de tramas de polling. Por esta razón, utilizamos otro temporizador (representado con un cuadrado azul) para controlar de manera exacta el número de pollings que se llevan a cabo. De esta manera sólo realizaremos un polling en el intervalo de tiempo que hay entre dos envíos de datos. Aún así, en el mejor de los casos, el número de tramas intercambiadas (6 tramas/dato) será mayor que en caso en el que se emplea sólo confirmación a nivel enlace (2 tramas/dato). Además, aumenta el overhead, tiempo de procesamiento, consumo, etc.

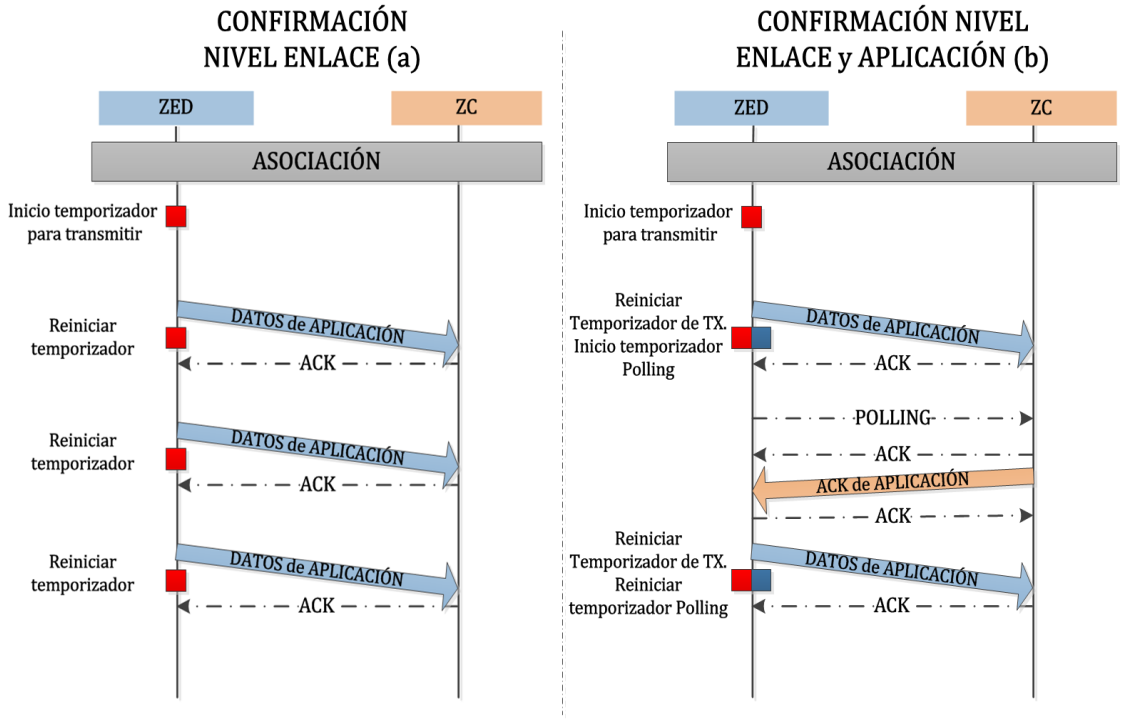

Figura 5-10: Estrategias de confirmación

Como se ha comentado en numerosas ocasiones, en el momento en que no se reciba una confirmación, ya sea de nivel enlace o aplicación, se producirá automáticamente una retransmisión de la trama.

Analizaremos a continuación el caso en el que el end-device utiliza un router para poder alcanzar al coordinador. Puesto que ahora disponemos de tres dispositivos colaborando para llevar a cabo una comunicación, podemos hablar de una red mesh. Aunque dispongamos de un número de dispositivos muy reducido, con estos ejemplos es posible observar algunas de las características más importantes de ZigBee como el multisalto o el descubrimiento de rutas. Como en el caso anterior, volveremos a estudiar los dos escenarios que plantean el tipo de confirmación que vayamos a utilizar.

5.2.8 - Red Mesh

En la Figura 5-11 se observa el caso en el que end-device no tiene cobertura suficiente para poder enviar directamente sus datos al coordinador y, por tanto, debe establecer una comunicación intermedia con el router. Como sucedía anteriormente, con la confirmación de nivel enlace se asegura que la trama ha llegado correctamente al siguiente salto.

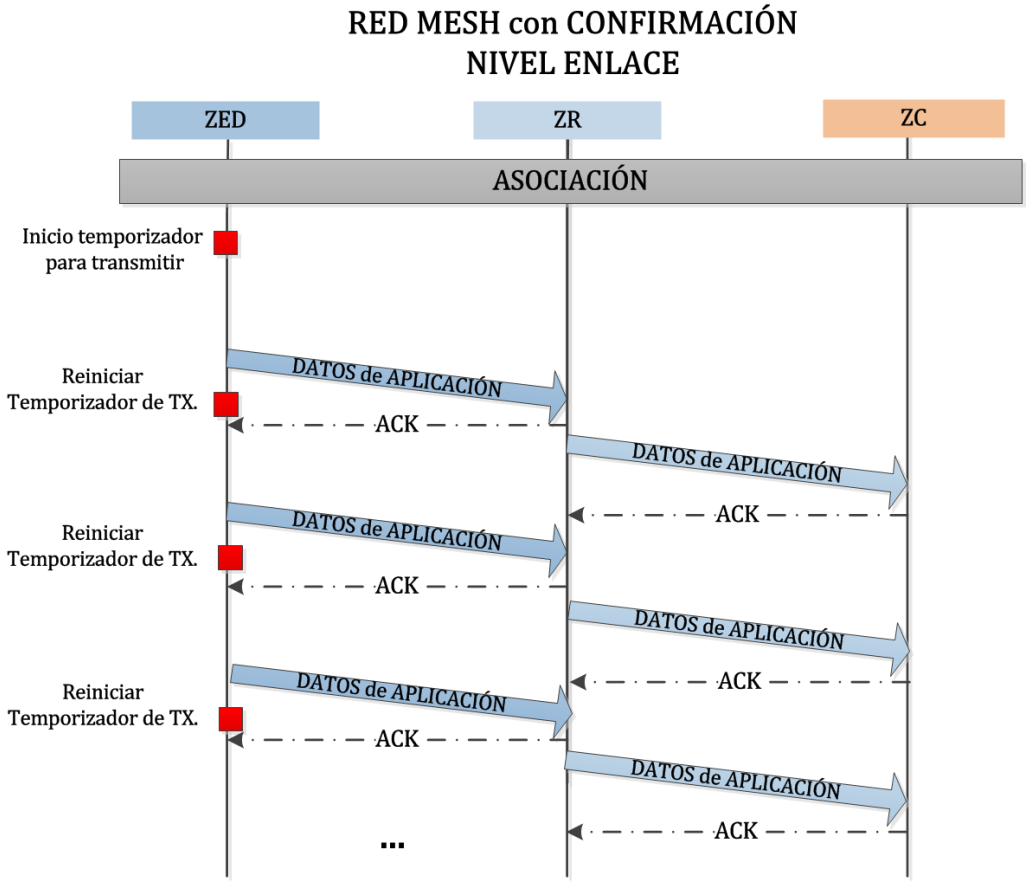

Figura 5-11: Red mesh con confirmación nivel enlace

No obstante, aunque la trama de datos sea recibida correctamente por el coordinador y todos los saltos hayan sido confirmados a nivel enlace, puede suceder que los datos de nivel aplicación no sean válidos. Por tanto, si empleamos esta estrategia deberemos tener en cuenta que no existe ninguna forma de avisar al end-device de que la información de nivel aplicación es errónea.

Como solución al problema anterior, donde el coordinador puede recibir datos de aplicación erróneos, planteamos utilizar confirmación de nivel aplicación. Ahora es posible notificar al end-device que los datos de aplicación enviados son correctos antes de que éste envíe la siguiente trama de datos. Como se puede apreciar Figura 5-12, será necesario un mayor número de tramas que en el caso anterior, ya que la confirmación de nivel aplicación requiere que el end-device emplee polling para recibir la confirmación.

Por tanto, deberemos tener en cuenta que en este escenario el end-device deberá estar mucho más tiempo activo que en el escenario anterior. Además, en el caso de que produzca una retransmisión de nivel aplicación, el end-device deberá volver a repetir todo el proceso de transmisión de datos y polling, con el consecuente gasto de tiempo y energía.

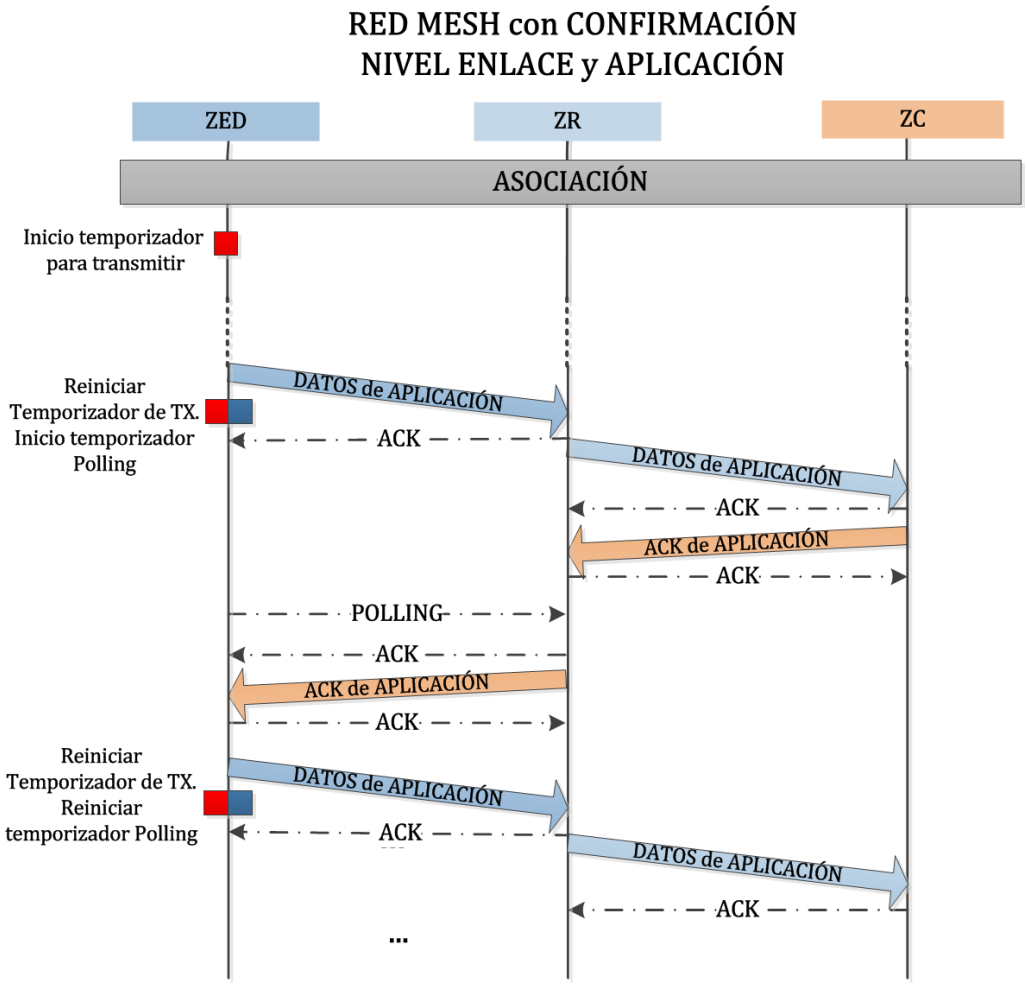

Figura 5-12: Red mesh con confirmación nivel enlace y aplicación

Después del análisis de estos escenarios podemos concluir que no existe una solución mejor que otra y que todas dependen del tipo de aplicación. Utilizar confirmación a nivel enlace asegura que los datos llegan al siguiente salto, pero no permite confirmar al end-device que los datos recibidos de nivel aplicación son correctos. Mientras que utilizar confirmación a nivel aplicación obliga a procesar un mayor número de tramas y a mantener al end-device más tiempo activo, con el consecuente consumo asociado.

Por tanto, será obligación del desarrollador estudiar qué tipo de confirmación y que topología es más adecuada para su aplicación.

Capítulo 6: EJEMPLOS DE APLICACIÓN

A lo largo de este capítulo se mostrará el funcionamiento de los escenarios de aplicación propuestos: comunicación directa y red mesh. Para ello, implementaremos un coordinador, un router y un end-device en diferentes placas (sección 4.4). A continuación, procederemos a capturar las tramas que se intercambian durante el proceso de comunicación utilizando la placa de captura (sniffer) CC2531. Mediante el uso de las capturas mostradas en el programa Packet Sniffer, analizaremos el intercambio de tramas y explicaremos los conceptos más importantes de la comunicación. A partir de los resultados obtenidos, se expondrán las ventajas e inconvenientes del uso de la tecnología ZigBee.

En el capítulo anterior se analizaron posibles estrategias para implementar una aplicación sencilla en la que un end-device envía datos de forma periódica a un coordinador. Además, se planteó la influencia que tiene en el sistema emplear confirmación a distintos niveles: enlace y aplicación. Para demostrar el correcto funcionamiento del estándar en las placas de SAYME, se desplegaron las redes representadas en la Figura 6-1, analizadas en el capítulo anterior. De modo que se comprobaron cuestiones como la creación de la red, el proceso de asociación, envío de mensajes, etc. En todos los experimentos se utilizó siempre confirmación a nivel aplicación debido a que resulta obligatorio en determinados profiles comerciales.

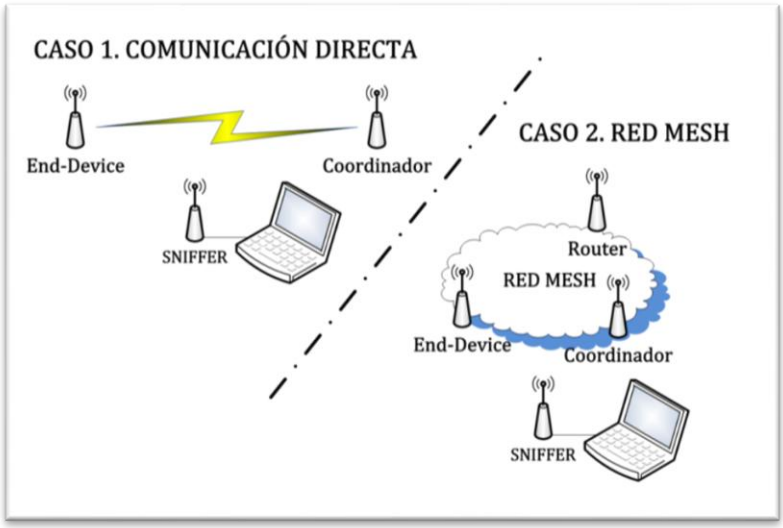

Figura 6-1: Casos de aplicación

6.1 - Comunicación directa

Como primer ejemplo de aplicación, se comprobó el funcionamiento propuesto en la Figura 5-10b. Para ello, se emplearon dos placas de SAYME (Figura 4-6) programadas para operar como end-device una y como coordinador la otra. En la Tabla 6-1 se recogen las direcciones MAC y red (NWK) asignadas a dichos dispositivos. La dirección de red del coordinador siempre es por defecto 0x0000, mientras que la del end-device es asignada de manera aleatoria por el coordinador cuando se asocia a la red.

Módulo	Dirección MAC	Dirección NWK
End-device	0xF380 1B12 5731 1045	0x089C
Coordinador	0x55AA ABCD 3FFF 55AA	0x0000

Tabla 6-1: Direcciones de los módulos

El identificador de la PAN es generado también por el coordinador de forma aleatoria y toma el valor de 0x6D9B en nuestro caso. Por otra parte, hemos elegido el canal 0x16 (2460 MHz) para llevar a cabo las comunicaciones. A continuación, detallaremos el intercambio de tramas que tiene lugar en nuestra aplicación.

Como se describió en Figura 5-7, los primeros pasos para la formación de una red son la búsqueda de una red y el proceso de asociación. En la Figura 6-2 podemos observar cómo se lleva a cabo la búsqueda de una red en el Z-Stack. Tal y como comentamos anteriormente, el end-device envía tramas de petición (beacon request) durante el escaneado activo para buscar una red a la que poder asociarse. Las tramas se envían en broadcast a todas las redes que puedan existir en el canal

0x16, elegido arbitrariamente por nosotros. Cuando el coordinador de una red reciba una de estas tramas, enviará al end-device una trama con información para que éste sepa cuáles son las características de la red y si permite la asociación.

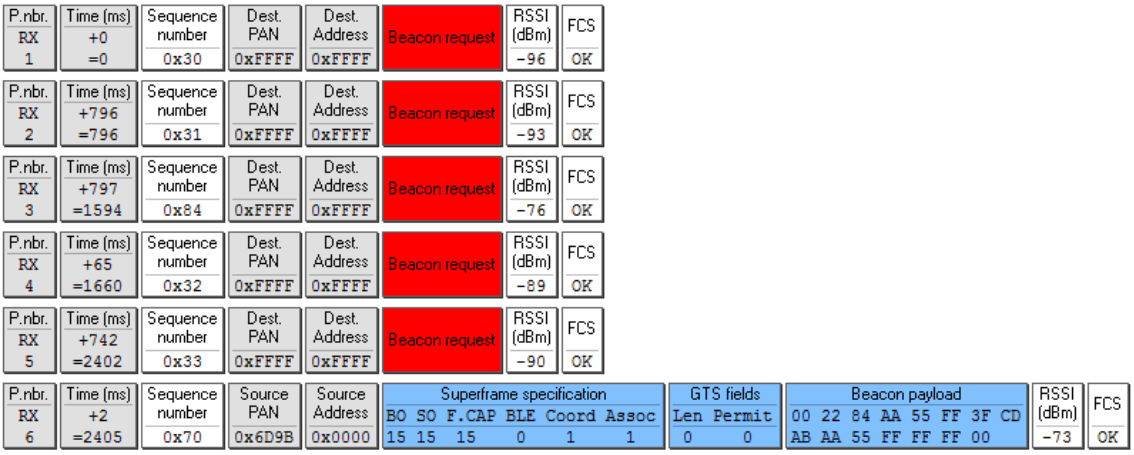

Figura 6-2: Búsqueda de una red

Una vez que el end-device reciba la trama con la información de la red, éste procederá a intentar conectarse a ella si es posible. Para ello, enviará una trama de petición de asociación (association request) a la dirección de red del coordinador, incluyendo su dirección MAC completa y el identificador de la PAN. Una vez recibida la petición, el coordinador responderá automáticamente con una trama de confirmación a nivel de enlace.

A continuación, el end-device comprobará si existe alguna información pendiente para él, enviando una trama de petición de datos (polling). Nuevamente, el coordinador confirmará que ha recibido la petición de datos con un ACK de nivel de enlace.

Por último, el coordinador envía al end-device una trama de confirmación a la petición de asociación donde se incluye la dirección NWK asignada al end-device. Como se aprecia en la Figura 6-3, el proceso de asociación concluye cuando el end-device confirma con un ACK la correcta recepción de la trama. A partir de este momento, el end-device forma parte de la red y podrá emplear su dirección NWK para comunicarse con cualquier otro dispositivo de la misma red. Como se puede observar, el proceso completo se ha realizado en 3.41 segundos.

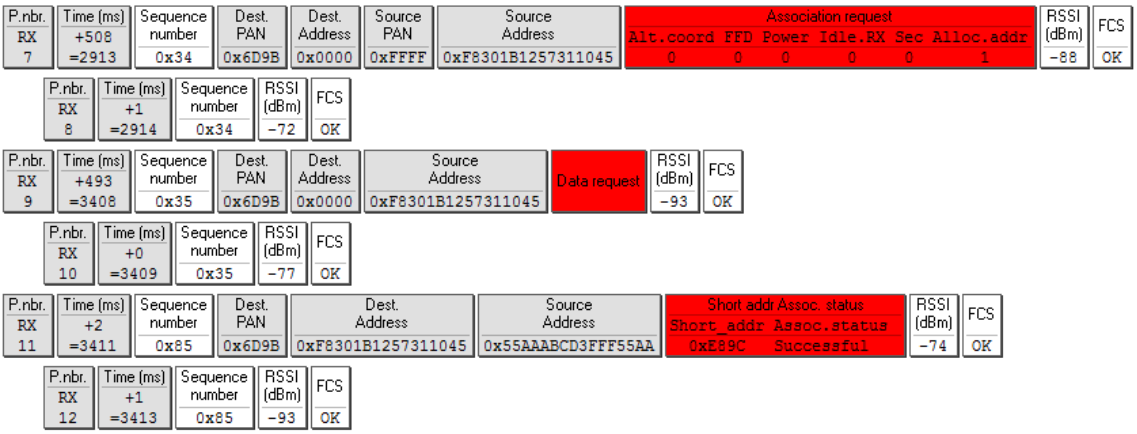

Figura 6-3: Proceso de asociación

Una vez concluido el proceso de asociación, empieza el funcionamiento real de nuestra aplicación. Como se explicó en la Figura 5-10b, el end-device envía periódicamente datos, en nuestro caso una cadena de caracteres, al coordinador de la red. Cuando el coordinador reciba los datos, enviará automáticamente un ACK de nivel de enlace para confirmar la correcta recepción. Tal

y como está diseñada la aplicación, el end-device requiere de una confirmación de aplicación para poder enviar el siguiente dato. Para ello, enviará una petición de datos (polling) después de la confirmación de enlace recibida. Nuevamente, el coordinador responderá con una confirmación de nivel enlace (a la trama de polling) y enviará después el ACK de nivel aplicación (APS Payload = 00) al end-device; éste lo recibirá y le contestará con un ACK de enlace al coordinador. A partir de este momento, todo el proceso se repetirá periódicamente en el tiempo hasta que el end-device abandone la red.

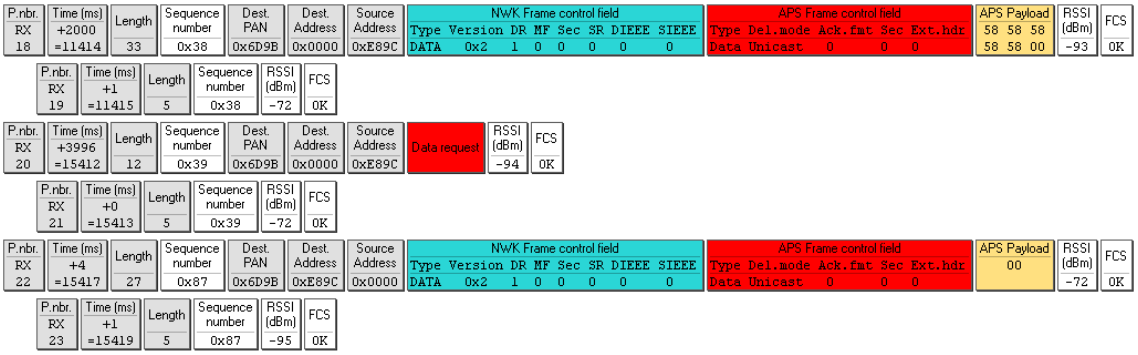

Figura 6-4: Envío de datos+Polling

A continuación, procederemos a explicar los resultados obtenidos en el escenario en el que implementamos una red mesh.

6.2 - Red Mesh

Para demostrar las ventajas que proporciona la red mesh, se implementó la aplicación descrita en la Figura 5-12. Para ello, se desplegó una red formada por un end-device que, por razones de cobertura, utiliza un router intermedio para poder transmitir sus datos al coordinador de la red.

Como se observó en la Figura 4-6, las placas utilizadas en el proyecto pueden emplear adicionalmente una antena externa para conseguir una mayor cobertura. Así pues, se instaló una antena en el router para que su cobertura alcanzase conjuntamente al end-device y al coordinador.

Una vez inicializada la red por el coordinador, se procedió a encender el router y se observó su proceso de asociación a la red. Después, se repitió el mismo procedimiento con el end-device. Tras un periodo de tiempo, el end-device encontró la red mediante el escaneado activo y a continuación recibió dos tramas con información de la misma red: una del router y otra del coordinador. Las tramas eran iguales (PAN ID, Stack ID, versión ZigBee, etc.) salvo el valor Coord (que indica quien es el coordiandor) y el campo de calidad del enlace (LQI). Como era lógico, debido a la proximidad que tenía el end-device con el router, el valor LQI era mucho mayor que el de la trama enviada por el coordinador. Por tanto, el end-device eligió como nodo padre al router para asociarse. Cuando la red ya estaba totalmente constituida, se procedió a aumentar la distancia entre el end-device y el coordinador para evitar una posible asociación de estos.

Tras un periodo de tiempo, se procedió a desconectar el router para comprobar los efectos que se producían en la red. Se observó que tras 8 envíos de datos sin recibir ningún ACK de enlace por parte del router, el end-device envió un mensaje de notificación de orfandad (Orphan notification) para indicar la necesidad de un nuevo dispositivo al que asociarse. Como se aprecia en la Figura 6-5, esta trama se envía de forma broadcast a todas las redes que pudieran existir en el canal.

P.nbr.	Time (ms)	Length	Frame control field					Sequence number	Dest. PAN	Dest. Address	Source Address		RSSI (dBm)	FCS
RX	+5		Type	Sec	Pnd	Ack.req	PAN_compr							
198	=118950	18	CMD	0	0	0	1	0x67	0xFFFF	0xFFFF	0xF8302E226402464C	Orphan notification	-90	OK

Figura 6-5: Notificación de orfandad

Una vez enviada la trama de notificación de orfandad, se observó que el end-device no era capaz de detectar al coordinador debido a que habíamos aumentado la distancia tras el proceso de asociación. Por tanto, se procedió a acercarlos para observar que efectos se producían. Como era evidente, el end-device detectó al coordinador y comenzó nuevamente el proceso de asociación a la red, esta vez con el coordinador como nodo padre. Una vez constituida la red, la aplicación volvió a funcionar siguiendo el esquema de comunicación directa explicado en la sección anterior (comunicación directa).

6.3 - Conclusiones

Con estos ejemplos se ha pretendido mostrar los beneficios más importantes que motivaron la utilización del estándar ZigBee como actualización del protocolo SWAN. Algunas de las características que hemos observado en los ejemplos prácticos son:

- **Fiabilidad y robustez**.
  - La topología mesh permite la auto-reparación de la red en situaciones no favorables, para ello actualiza dinámicamente las conexiones entre dispositivos. Si un dispositivo abandona la red, sus vecinos pueden encontrar fácilmente una ruta alternativa para encaminar los datos.
  - Mediante la auto-configuración, los dispositivos pueden unirse automáticamente a la red. El coordinador detecta cuando un dispositivo quiere unirse, identifica que tipo de dispositivo es (router o end-device), dónde están sus vecinos y cuál es la mejor ruta para llegar a él. Cuando un dispositivo empieza a formar parte de una red, éste mantendrá actualizadas las rutas teniendo en cuenta los posibles cambios en la calidad de los enlaces.
  - En las redes mesh, la comunicación siempre se realizará de forma que se alcance únicamente al nodo más cercano, de modo que la interferencia con otras señales es muchísimo menor.
  - Podemos emplear diferentes niveles de confirmación para verificar la correcta recepción de los datos, aumentando, por tanto, la fiabilidad en la información.

- **Flexibilidad y escalabilidad**.
  - Las técnicas de encaminamiento proporcionan gran flexibilidad al sistema, ya que permiten generar rutas desde cualquier nodo emisor a cualquier nodo destino de la red.
  - La red mesh descubre automáticamente los nuevos nodos que quieran incorporarse a la red. Por tanto, es posible aumentar el rango de cobertura, añadir redundancias y mejorar la calidad del enlace, simplemente añadiendo nuevos nodos.

Sin embargo, todas estas prestaciones adicionales que proporciona ZigBee tienen una serie de costes que han de ser considerados en el desarrollo de las aplicaciones:

- **Mayor overhead en las tramas**. Implica un mayor consumo energético y un incremento en el tiempo de procesamiento.
- Las **prestaciones de nivel de red** requieren que el microcontrolador esté más tiempo activo que en SWAN, con el consecuente coste energético.
- **Control energético automatizado por el OSAL**. El desarrollador no puede decidir cuando el end-device duerme. Por tanto, se dificulta en gran medida el diseño de aplicaciones de ultra-bajo consumo.
- El OSAL gestiona numerosos **procesos automatizados** (descubrimiento de vecinos, mantenimiento de rutas, comprobación de redes vecinas, etc.) en el dispositivo. El microcontrolador tiene que estar mucho más tiempo activo a la espera de posibles mensajes.

A la vista de estos resultados, se puede concluir que utilizar el estándar ZigBee proporciona numerosas ventajas adicionales para el desarrollo de redes de sensores inalámbricas. No obstante, todas estas prestaciones aumentan el consumo energético respecto al protocolo SWAN. Puesto que es posible implementar en el mismo hardware ambos protocolos, será tarea del desarrollador determinar que aplicaciones o entornos son más favorables para cada uno de ellos.

Capítulo 7: CONCLUSIONES Y LÍNEAS FUTURAS

7.1 - Conclusiones

Al comienzo del presente proyecto se proponía estudiar el novedoso estándar 6LoWPAN como una alternativa real al protocolo de comunicaciones SWAN, utilizado por la compañía SAYME en su plataforma SENSbee para redes de sensores inalámbricas. La idea original del proyecto era analizar las ventajas que podría reportar el hecho de implementar una tecnología estandarizada en una plataforma comercial para redes de sensores. Como objetivo adicional, se pretendía además aprovechar los módulos de la plataforma SENSbee para intentar integrar el estándar 6LoWPAN.

Lamentablemente tras el estudio realizado se concluyó que, aun siendo un protocolo con muchísimo futuro (no olvidemos que es una adaptación de IPv6 para redes de sensores) todavía está muy lejos de ser una alternativa real ya que se encuentra en fase de desarrollo. Además, LoWPAN está orientado a dispositivos más complejos que los que se emplean en las redes de sensores inalámbricas. Por tanto, rápidamente se descartó como alternativa y se procedió a buscar otra tecnología que aportará nuevas prestaciones con el valor añadido de que fuera estándar.

Así pues, se sugirió como alternativa el estándar ZigBee. Al igual que 6LoWPAN, ZigBee es un estándar para redes de sensores inalámbricas que presenta características muy interesantes en el desarrollo de aplicaciones de monitorización y control remoto. El proyecto continuó con el estudio de las prestaciones que podía proporcionar ZigBee, concluyendo finalmente con su elección.

Como comentamos anteriormente, la integración del estándar era uno de los objetivos principales del presente proyecto. Así pues, se eligió el stack oficial de Texas Instruments debido a que está diseñado específicamente para la misma familia de microcontroladores y chip de comunicaciones que los utilizados por SAYME en sus módulos. Tras un exhaustivo estudio del stack y de sus requisitos, se determinó que era necesario cambiar a una versión más actualizada del microcontrolador debido a los altos requisitos de memoria. Por otro lado, fue necesario un rediseño completo de la interfaz de gestión del chip de comunicaciones, fabricándose finalmente un nuevo modelo de placa.

Una vez recibidos los nuevos modelos, se procedió a desarrollar una serie de aplicaciones que comprobasen la correcta integración del estándar y que mostrasen algunas de las prestaciones más importantes que aporta ZigBee. A la vista de los resultados obtenidos, podemos concluir que se alcanzaron los objetivos iniciales de buscar un estándar para redes de sensores que aportará nuevas funcionalidades, que pudiésemos implementarlo en el mismo hardware de forma satisfactoria y que tuviese el valor añadido de ser un estándar.

Por otra parte, este proyecto ha supuesto una experiencia muy enriquecedora para mí, tanto en lo personal como en lo académico. En primer lugar, me ha permitido descubrir el ámbito de las redes de sensores que desconocía completamente, dándome la oportunidad de adquirir nuevos conocimientos que han complementado en gran medida mi formación.

En segundo lugar, poder participar en un "proyecto de ingeniera" ha sido una experiencia muy satisfactoria. He tratado con las dificultades que habitualmente pueden surgir cuando se realizan este tipo de proyectos y he tenido que buscar soluciones para completarlo. Además, haber podido participar de manera directa en algunas de las decisiones tomadas con el equipo de SAYME ha sido una experiencia enriquecedora, que me ha permitido aprender cómo funciona un proyecto en el mundo real, descubriendo las numerosas dificultades que presenta desarrollar un producto tecnológico.

Por último, la posibilidad de implementar físicamente una red ZigBee ha sido todo un logro personal. Para ello, he tenido que aprender a manejar un hardware nuevo para mí, programar por primera vez un sistema en tiempo real y utilizar un stack comercial de ZigBee.

Por tanto, la realización de este proyecto me ha ayudado a complementar todo el conocimiento teórico adquirido durante mis años de Universidad.

7.2 - Líneas futuras

En cuanto a las líneas futuras, considero que sería muy interesante observar cómo evoluciona 6LoWPAN, puesto que seguramente en unos años se convierta en una tecnología muy importante en el mercado de las redes de sensores inalámbricas.

En cuanto a ZigBee, sería muy interesante estudiar en profundidad el funcionamiento del OSAL del Z-Stack, para poder entender con más detalle cómo funciona la implementación de Texas Instruments. Por ejemplo, se podría comprender mejor como se realiza la gestión energética de los dispositivos para poder mejorarla si fuera posible y desarrollar, por tanto, aplicaciones con una gestión más controlada del consumo.

Además, se podrían investigar las funcionalidades de los profiles oficiales y qué prestaciones reportarían en el desarrollo específico de aplicaciones comerciales.

Por último, también sería muy importante conocer los beneficios que puede reportar la certificación de una plataforma propietaria en la ZigBee Alliance y qué requisitos de interoperabilidad habría que cumplir.

Capítulo 8: REFERENCIAS

[1] P. T. Alonso, Redes de sensores: Fundamentos y aplicaciones, Santander, Julio 2008.

[2] F. R. Pascual, Redes de sensores inalámbricos, Valencia.

[3] SAYME, «http://sayme.es/que-es-sensbee/tecnologia,».

[4] IEEE 802.15.4 Specification: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low Rate Wireless Personal Area Networks (LR-WPANS), 2005.

[5] Home Networking with IEEE 802.15.4: A Developing Standard for Low-Rate. A Developing Standard for Low-Rate, 2002.

[6] D. Stevanovic, ZigBee/ IEEE 802.15.4 Standard, Junio 2007.

[7] D. W. Naveen Sastry, Security Considerations for IEEE 802.15.4 Networks, University of California, Berkeley, 2004.

[8] F. Moreno, Monitorización inalámbrica de grúas torre mediante tecnología ZigBee, Universidad de Cantabria, 2007.

[9] M. J. S. Díaz, Desarrollo de un repetidor para la red de sensores SAYME basada en el estándar IEEE 802.15.4, Universidad de Cantabria, 2007.

[10] MSP430F261x, MSP430F241x Mixed Signal Microcontroller Datasheets, Texas Instruments.

[11] C. B. Zach Shelby, 6LoWPAN: The Wireless Embedded Internet, Wiley, 2010.

[12] Sub-1GHz 6LoWPAN Development Kit, Texas Instruments, 2011.

[13] 6LoWPAN APIs User Guide, Jennic, 2010.

[14] ZigBee Specification: ZigBee Document 053474r17, ZigBee Alliance, Junio 2008.

[15] CC2520 Datasheets, Texas Instruments.

[16] IAR, «http://www.iar.com,».

[17] CC2531 USB Hardware User's Guide, Texas Instruments.

[18] Z-Stack: Application Programming Interface, Texas Instruments.

[19] Power Management For MSP430 and CC2520 Radio, Texas Instruments.

[20] D. Gislason, ZigBee Wireless Networking, Primera edición ed., Agosto 2008.