

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



**Proyecto Fin de Carrera**

## **Una técnica para la caracterización de nodos en Redes de Sensores Inalámbricas**



AUTOR: Javier Alfonso Avilés García  
DIRECTORES: Fernando Losilla López  
Iain Phillips

Julio / 2008

<b>Autor</b>	Javier Alfonso Avilés García
<b>E-mail del Autor</b>	<a href="mailto:javiles.garcia@hotmail.com">javiles.garcia@hotmail.com</a>
<b>Director(es)</b>	Fernando Losilla López / Iain Phillips (Universidad de Loughborough, Reino Unido)
<b>E-mail del Director</b>	<a href="mailto:fernando.losilla@upct.es">fernando.losilla@upct.es</a>
<b>Codirector(es)</b>	
<b>Título del PFC</b>	Una técnica para la caracterización de nodos en Redes de Sensores Inalámbricas.
<b>Descriptor(es)</b>	sensor, WSN, caracterización, metodología
<b>Resumen</b>	
<p>Actualmente se pueden adquirir multitud de juegos de sensores comerciales con el objetivo de desplegar una Red de Sensores Inalámbrica. Sin embargo, normalmente las hojas de características de estos dispositivos no contienen las especificaciones suficientes que permiten al diseñador de la red conocer cual va a ser su comportamiento una vez estén desplegados en el sistema real. Por ello, en este trabajo se ha desarrollado una metodología, constituida por tres experimentos diferentes, que permite caracterizar cualquier conjunto de sensores con el objetivo de descubrir como se comportarían. Antes de la descripción de dicha metodología se realiza una introducción a las Redes de Sensores Inalámbricas. Además, se ha comprobado el correcto diseño de la técnica de caracterización con su ejecución sobre un juego de sensores de la compañía <i>Sensinode</i>. Finalmente, tras completar los experimentos, los datos obtenidos han sido analizados y se han extraído conclusiones sobre los hallazgos encontrados.</p>	
<b>Titulación</b>	Ingeniería de Telecomunicación
<b>Intensificación</b>	
<b>Departamento</b>	Tecnologías de la Información y las Comunicaciones
<b>Fecha de Presentación</b>	Julio - 2008

*A mis padres y hermanos*

*A mis amigos*

*A mis directores de proyecto*

**GRACIAS**



# Índice

---

Índice .....	i
Índice de figuras .....	iii
Índice de tablas .....	v
<b>Introducción .....</b>	<b>1</b>
1. Motivación y objetivos.....	1
2. Trabajos relacionados .....	4
<b>Capítulo 1. Redes de Sensores Inalámbricas .....</b>	<b>7</b>
1.1 Definición.....	7
1.2 Características.....	10
1.3 Aplicaciones.....	17
1.3.1 Aplicaciones militares .....	17
1.3.2 Aplicaciones medioambientales.....	¡Error! Marcador no definido.8
1.3.3 Aplicaciones sanitarias .....	¡Error! Marcador no definido.9
1.3.4 Aplicaciones de hogar: domótica .....	¡Error! Marcador no definido.0
1.3.5 Otras aplicaciones comerciales .....	¡Error! Marcador no definido.0
1.4 Principales motes comerciales .....	21
1.4.1 Sensinode series .....	24
1.4.2 Telos .....	29
1.4.3 Micas .....	3¡Error! Marcador no definido.
1.4.4 Comparación entre tecnologías .....	35
<b>Capítulo 2. Metodología para la caracterización de nodos en WSNs .....</b>	<b>39</b>
2.1 Introducción .....	39
2.2 Variables experimentales .....	40
2.2.1 Distancia .....	41
2.2.2 Potencia de transmisión .....	43
2.2.3 Localización .....	43
2.2.4 Distribución del tráfico de paquetes.....	46
2.2.5 Protocolo de la capa de transporte .....	47
2.2.6 Número de nodos.....	48
2.3 Experimento A .....	49
2.3.1 Tasa de recepción de paquetes.....	50

2.3.2 Fuerza de la Señal Recibida.....	52
2.3.3 Indicador de Calidad del Enlace .....	53
2.3.4 Procedimiento.....	54
2.4 Experimento B .....	57
2.4.1 Orientación de los nodos .....	57
2.4.2 Procedimiento.....	58
2.5 Experimento C .....	60
2.5.1 Vida de la batería .....	61
2.5.2 Procedimiento.....	63
<b>Capítulo 3. Caracterización de un nodo comercial .....</b>	<b>65</b>
3.1 Sensor Micro.2420 U100 .....	65
3.2 D100 Devboard.....	69
3.3 Ejecución del Experimento A.....	71
3.3.1 Descripción del software.....	71
3.3.2 Resultados y análisis .....	73
3.4 Ejecución del Experimento B.....	80
3.4.1 Descripción del software.....	80
3.4.2 Resultados y análisis .....	80
3.5 Ejecución del Experimento C.....	83
3.5.1 Descripción del software.....	84
3.5.2 Resultados y análisis .....	86
<b>Capítulo 4. Conclusiones.....</b>	<b>89</b>
<b>Anexo A.....</b>	<b>91</b>
<b>Anexo B.....</b>	<b>105</b>
<b>Referencias .....</b>	<b>123</b>

# Índice de figuras

---

Fig 1.1 Red de Sensores Inalámbrica .....	7
Fig 1.2 Elementos de una WSN .....	11
Fig 1.3 Modelo de trabajo de IEEE 802.15.4 y ZigBee.....	11
Fig 1.4 Arquitectura centralizada de una WSN.....	13
Fig 1.5 Arquitectura distribuida de una WSN.....	14
Fig 1.6 Chip para el estudio del cuerpo humano .....	19
Fig 1.7 Estructura hardware básica de un sensor inalámbrico inteligente .....	21
Fig 1.8 Ejemplos de unidades sensoras.....	22
Fig 1.9 Consumo de un nodo sensor según su estado .....	23
Fig 1.10 Arquitectura de NanoStack.....	24
Fig 1.11 Componentes internos de la arquitectura NanoStack .....	25
Fig 1.12 Detalle de los componentes internos de NanoStack .....	26
Fig 1.13 NanoSensor N711 de las Nano series de <i>Sensinode</i> .....	28
Fig 1.14 Sensor Micro.2420 U100 de las Micro series de <i>Sensinode</i> .....	28
Fig 1.15 Modelo Telos TPR2420.....	29
Fig 1.16 Diagrama de bloques del modelo TPR2420.....	31
Fig 1.17 Placa de desarrollo MIB510.....	32
Fig 1.18 MicaZ.....	32
Fig 1.19 Mica2 .....	33
Fig 1.20 Diagrama de bloques del modelo MPR400 de Mica2 .....	34
Fig 1.21 Mica2Dot.....	34
Fig 1.22 Diagrama de bloques del modelo MPR500 de Mica2Dot.....	35
Fig 1.23 Protocolos Zigbee y 6LoWPAN comparados .....	37
Fig 2.1 Pasillo donde se han ejecutado los experimentos.....	45
Fig 2.2 Parcela de césped donde se han ejecutado los experimentos .....	45
Fig 2.3 Parking de asfalto donde se han ejecutado los experimentos .....	46
Fig 2.4 Disposición de los elementos en las distintas localizaciones .....	56
Fig 2.5 Ejemplo de colocación de los nodos sensores para el Experimento B .....	60
Fig 2.6 Relación entre la vida de la batería NiMH y su voltaje .....	62
Fig 2.7 Posición de los nodos sensores en el Experimento C.....	64
Fig 3.1 Dispositivo Micro.2420 U100 .....	65
Fig 3.2 Visualización de la antena chip Fractus Slim Reach Xtend.....	67
Fig 3.3 Diagrama de radiación de la antena chip Fractus Slim Reach Xtend .....	68

Fig 3.4 Diagrama de bloques del dispositivo D100 Devboard.....	69
Fig 3.5 Micro.2420 junto con módulo A500 conectados a la placa D100 Devboard ....	70
Fig 3.6 Relación entre la tasa de recepción, la distancia y la potencia de emisión.....	74
Fig 3.7 Mapa de configuración para optimizar la energía y la tasa de recepción.....	75
Fig 3.8 Relación entre el RSS, la distancia y la potencia de transmisión.....	77
Fig 3.9 Obtención del RSS mínimo .....	78
Fig 3.10 Relación entre el LQI, la distancia y la potencia de transmisión .....	80
Fig 3.11 Relación entre tasa de recepción y distancia para cada posición del emisor	82
Fig 3.12 Esquema de la comunicación entre nodos en el Experimento B .....	83
Fig 3.13 Nivel de batería en relación a los mensajes enviados/recibidos y al tiempo..	87



# Índice de tablas

---

Tabla 1.1 Productos de la Micro series de Sensinode .....	29
Tabla 1.2 Resumen comparativo de distintos motes.....	36
Tabla 2.1 Valores de los factores para cada experimento .....	49

# Introducción

---

## 1. Motivación y objetivos

A pesar del gran desarrollo de la industria de las telecomunicaciones y la electrónica en las últimas décadas, la comunicación inalámbrica entre dispositivos sigue siendo aún muy impredecible. El grado de incertidumbre es todavía mayor si nos adentramos en el campo de las Redes de Sensores Inalámbricas (Wireless Sensor Networks, WSNs). El hecho de que estas entidades sean normalmente diseñadas para monitorizar y ejecutar aplicaciones de carácter específico (redes ad-hoc), unido a la carencia de estándares comunes entre tecnologías, así como de herramientas para la simulación, intensifica y convierte el desarrollo y despliegue de este tipo de redes en una tarea más que compleja.

Actualmente, cualquier usuario o empresa interesada en desplegar una WSN, puede adquirir multitud de juegos de sensores fabricados por muchas firmas diferentes. Sin embargo, una cantidad importante de estas colecciones de dispositivos no son capaces de comunicarse entre si, ya que en ocasiones sus características y protocolos de comunicación son distintos.

A pesar de que los nodos que constituyen una red inalámbrica son ubicados normalmente en diferentes localizaciones para llevar a cabo aplicaciones particulares, las hojas de características de estos dispositivos no especifican como será su comportamiento para cada una de las ubicaciones y condiciones, y por tanto es necesario que sean testeados previamente sobre el escenario real en el que serán desplegados. Como resultado de este hecho, es condición necesaria la existencia de una metodología que permita al creador de la WSN disponer de una guía para ejecutar un procedimiento de testeo de los sensores que se tiene previsto se van a utilizar.

Aunque todo este proceso de prueba y experimentación con los sensores sobre el escenario real suponga un alto esfuerzo mecánico y económico, representa el camino más viable para que el diseñador de la red disponga de una descripción correcta y amplia del comportamiento verdadero de los dispositivos en dicho escenario. Así por ejemplo, los procedimientos de simulación existentes en el campo de las WSNs, la mayoría de las veces no es válido su uso, ya que proporcionan resultados que tienen

dependencia sobre varias suposiciones teóricas, las cuales no son suficientemente precisas para representar el verdadero comportamiento de la red en el escenario real.

Cuando la metodología para el testeo de los sensores de una red es completada, el usuario recibe información, la cual tras ser estudiada y analizada, le permite conocer las características y el comportamiento de los dispositivos. Todo este proceso es llamado *caracterización* de un juego de sensores inalámbricos. Ésta facilitará al diseñador la correcta toma de decisiones a la hora de desplegar su red para que las especificaciones y requerimientos (criterios de calidad) del sistema se satisfagan. Véase por medio de un par de ejemplos:

- Por un lado, se desea desplegar en un área específica una WSN con el fin de monitorizar diferentes variables climáticas. Tras consultar distintos juegos de sensores comerciales, ninguno especifica como se comportarían los dispositivos en el escenario real de montaje de la red. Por tanto, se decide adquirir un número reducido de sensores de la firma A para llevar a cabo su caracterización. Al principio se debería comprar el mínimo número posible de dispositivos, con el fin de que la repercusión sobre el presupuesto sea la menor posible una vez se diera el caso de que dichos sensores no fueran adecuados para su utilización en la red. Así pues, tras la caracterización, se averigua que la comunicación en el entorno es bastante pobre, ya que el alcance máximo entre dos dispositivos cualquiera es muy corto a la máxima potencia de transmisión. Además la duración de la vida de sus baterías es bastante escasa. Por consiguiente, si se dispusiera de un alto presupuesto, se podría hacer frente al proyecto de despliegue de la red, adquiriendo una gran suma de dispositivos y colocándolos en posiciones muy cercanas los unos a los otros (la más alta distancia de separación sería el alcance máximo), y configurados para transmitir a la mínima potencia, con el fin de ahorrar la mayor cantidad de energía posible. Sin embargo, si no se dispusiera del capital suficiente para adquirir el número necesario de nodos, deberían testearse otros juegos de sensores o abandonar el proyecto.
- Por otro lado, para desplegar la misma WSN en el mismo escenario, se compran sensores de la firma B, que tras ser caracterizados se obtiene que el alcance máximo en la comunicación entre dispositivos es bastante alto, y la vida de sus baterías es significativamente longeva. De este modo, en principio se podría disponer de un bajo presupuesto para afrontar el despliegue de la

red, ya que muy pocos sensores inalámbricos serían necesarios. Esto es porque las distancias de separación entre nodos podrían ser muy grandes (y por tanto se necesitan menos sensores para ocupar el espacio) debido al gran alcance en la comunicación y a que se podría trabajar con potencias de transmisión relativamente elevadas.

Como conclusión, es vital llevar a cabo la caracterización de la colección de sensores que constituirán una WSN antes de que los dispositivos sean desplegados, para asegurarse que el sistema que ellos formarán, cumplirá las especificaciones y requerimientos solicitados. Incluso, si dicha caracterización sobre los mismos sensores fue realizada para proveer otra red inalámbrica previamente, ella debe volver a repetirse, ya que las características del nuevo sistema pueden ser diferentes.

De este modo, los objetivos a conseguir con este proyecto son:

- Ampliar los conocimientos teóricos sobre las Redes de Sensores Inalámbricas (WSNs).
- Conocer detalladamente diferentes tecnologías de motes, empleados en la constitución de WSNs.
- Diseño de una metodología que permita el testeo de las características y comportamiento de cualquier juego comercial de sensores inalámbricos.
- Ejecutar dicha metodología sobre un específico juego de sensores obtenible en el mercado.
- Saber analizar la información recogida tras la ejecución de la metodología con el fin de llevar a cabo una correcta caracterización de los dispositivos con los que se ha experimentado.

## 2. Trabajos relacionados

Numerosos trabajos, donde se lleva a cabo la caracterización de sensores inalámbricos, han sido escritos en los últimos años. Sin embargo, sólo unos pocos completan un amplio test para estudiar el comportamiento de los dispositivos, así como sólo un reducido número de estos trabajos describen de forma detallada y extensa las técnicas sistemáticas que permiten al diseñador obtener la caracterización de sus sensores.

A modo de ejemplo, en 2006, varios investigadores, tal como S. Lin y J. Zhang [1], completaron numerosos experimentos sobre un conjunto de sensores inalámbricos, que les permitieron concluir que la calidad de la comunicación vía radio para esos nodos mostraba una gran inestabilidad según el entorno y el tiempo de ejecución de los tests. Al igual que será llevado a cabo en el presente trabajo, Lin y Zhang estudiaron profundamente varios parámetros del radioenlace, tales como la Fuerza de la Señal Recibida (Received Signal Strength, RSS) o el Indicador de Calidad del Enlace (Link Quality Indicator, LQI). Así, tras los resultados obtenidos, concluyeron que estos dos factores tenían una relación totalmente directa con la calidad del enlace, guiándoles tales observaciones a establecer un modelo para predecir la apropiada potencia de transmisión de los nodos que garantizara una buena tasa de recepción de paquetes. Por último, indicar que sin embargo, en su trabajo, al contrario que en el nuestro, no se detallan de forma minuciosa las técnicas para testear los dispositivos.

Más relacionado con el presente trabajo se encuentra un estudio publicado por J. Zhao y R. Govidan [2] en 2003 sobre los motes Mica [3], donde se investiga profundamente la recepción de paquetes en este tipo de sensores. En este caso, sí se describe ampliamente la metodología de testeo de los dispositivos, definiéndola como un primer paso hacia una colección de técnicas sistemáticas para realizar el estudio del comportamiento de redes de sensores en diferentes escenarios. Respecto a sus hallazgos, básicamente se explican los resultados obtenidos sobre la probabilidad de entrega y pérdida de paquetes en la capa física y la capa MAC (Medium Access Control). Además, al igual que en este proyecto, la experimentación es llevada a cabo en distintas ubicaciones, obteniendo resultados bastante dispares para cada una de ellas. También, se demuestra la existencia de “áreas grises”, donde se divisan irregularidades en la comunicación. Como explicación, se afirma que el fenómeno multi-camino es un posible responsable de este suceso.

Existen muchos otros trabajos donde una única característica de los sensores es investigada exclusivamente. La más común de ellas es la vida de la batería (battery life), la cual es probablemente el factor más importante de cualquier WSN. Por ejemplo, autores como S. Blom y C. Bellettini [4] describen en su estudio la relación entre el nivel de batería y la potencia emitida por los sensores inalámbricos EYESIFXv2 Infineon, creados dentro del proyecto europeo EYES [5]. De la misma forma que en el presente trabajo, aquí la configuración de los experimentos es detallada paso a paso. Así, las medidas son tomadas en un entorno cerrado (cualquier edificio donde las condiciones del tiempo no afecten los experimentos), donde cinco receptores, los cuales se encuentran conectados a diferentes ordenadores para el almacenamiento de datos, capturan mensajes de un único emisor. Finalmente, se concluye que la relación entre la potencia transmitida por los dispositivos y la vida de su batería es no lineal.

Así pues, a modo de resumen, la mayoría de los trabajos se preocupan más del análisis de resultados y la descripción de los hallazgos que de la propia definición de los procesos que permiten llevar a cabo los experimentos y conseguir las distintas colecciones de medidas. En cambio, en este proyecto, además de la síntesis y estudio de los resultados, se define minuciosamente todo el procedimiento para testear los dispositivos, a fin de que el usuario disponga de una herramienta que le permita experimentar con los diferentes juegos de sensores que ofrece el mercado.



# Capítulo 1

## Redes de Sensores Inalámbricas

---

### 1.1 Definición

Una Red de Sensores Inalámbrica (Wireless Sensor Network, WSN) es la colección de un conjunto de nodos sensores (motes) no cableados, de reducido tamaño, peso ligero y bajo coste, los cuales se despliegan en una región en particular (ver la Figura 1.1). Su función es proveer una infraestructura de comunicación inalámbrica para poder monitorizar cualquier sistema específico, normalmente mediante el control de parámetros físicos, tal como temperatura o presión. Además, al estar dotados con procesador, estos nodos son capaces de realizar ciertas computaciones locales sobre los datos tomados, lo que permite una reducción de tráfico a través de la red.

La construcción de estos nodos sensores ha sido posible debido a los grandes avances en el área de sistemas micro electromecánicos (MEMS) en los últimos años. Los MEMS son dispositivos en miniatura fabricados con capacidad de adquisición de datos, comunicación y procesamiento. Ellos se elaboran sobre un substrato de silicón, de la misma manera que los circuitos integrados, que combinan circuitería electrónica (sensores) con actuadores mecánicos, por ejemplo.



**Figura 1.1** Red de Sensores Inalámbrica.

Cada uno de estos nodos se divide en tres subsistemas principales:



- El subsistema sensor: capta el fenómeno existente en el medio ambiente.
- El subsistema de procesado: lleva a cabo el cómputo de la información recabada por los sensores.
- El subsistema de comunicación: es el responsable del intercambio de mensajes con el resto de entidades vecinas de la red.

Por otro lado, mientras que un único nodo posee su región de adquisición de datos, su potencia de procesamiento y una cantidad de energía limitada, la unión de un conjunto de ellos proporciona un aumento en la robustez, confiabilidad y precisión en el área de cobertura de dicha red. Esto ocurre ya que al tener más dispositivos que cooperan y colaboran entre sí en una región determinada, la adquisición de información es más “estrecha”, brindando así, más seguridad en la captación de fenómenos y reafirmando como consecuencia el buen desempeño de la red.

Dentro de una misma red se pueden tener diferentes tipos de sensores, los cuales monitorizan principalmente factores del medio ambiente, tal como los que se indicaron anteriormente. Sin embargo, pueden realizar muchas otras funciones, como por ejemplo realizar el seguimiento de ciertos objetos o entidades en el escenario de estudio mediante el uso de sensores que detectan el movimiento.

Actualmente, los campos donde las WSNs son empleadas son muy numerosos. Algunos de ellos son: agricultura inteligente y adquisición de datos en el medio ambiente, seguridad y aplicaciones militares, monitorización y control industrial, automatización del hogar, y aplicaciones médicas [6].

El proceso de diseño de una red inalámbrica de sensores involucra diversos retos, a los cuales se tiene que enfrentar su creador. Estos son: el tipo de arquitectura, el modo de disseminación y recolección de la información, la optimización en la localización de las diferentes entidades que forman la red y el aumento de la eficiencia en el consumo de energía. Este último es el factor primario que limita la vida útil de la red.

Normalmente, la actividad de medir parámetros, se da de manera periódica o esporádica. Tienen carácter periódico la mayoría de las mediciones de los fenómenos

captados en el medio ambiente, tal como la temperatura, la humedad o la radiación nuclear. El detectar la entrada de un intruso o la detección del estrés crítico de estructuras o maquinaria, son claros ejemplos de actividades esporádicas de adquisición de datos.

Por otro lado, existen diversas áreas que caracterizan a las WSNs frente a cualquier red inalámbrica convencional:

- Tamaño de la red: el número de dispositivos de una WSN suele ser más grande que el del resto de redes inalámbricas.
- Limitación de energía: en el caso de las redes de sensores, se espera trabajen en entornos agresivos en cuanto a las condiciones ambientales se refiere, y además con la mínima y a poder ser nula supervisión humana. Emplear este tipo de redes, donde la única fuente de alimentación para el nodo sensor es la batería, nos limita la vida útil del sistema, exigiendo de esta manera un conjunto de protocolos de red muy eficientes a nivel de la capa de red, de la capa de enlace de datos e incluso de la capa física, para brindar un control óptimo de la energía. A modo de resumen, los distintos tipos de fuentes de alimentación en WSNs se pueden clasificar de la siguiente manera:
  - Recargables.
  - No recargables
  - Regenerativas (capacidad de obtener energía a partir del parámetro físico de estudio).
- Distribución del tráfico: el patrón de tráfico varía en base al tipo de aplicación de la red. La adquisición de datos en un factor ambiental, normalmente genera de manera periódica la transmisión de pequeños paquetes a una estación central, la cual los analiza. En general, este tráfico demanda un bajo ancho de banda. Sin embargo, para otros tipos de aplicaciones, como el control de incendios, la información es enviada acorde a algún evento que ocurre en el medio, pudiendo ser el tráfico, acorde al tipo de sistema, más o menos denso que en los de carácter periódico.

## 1.2 Características

Cada una de los elementos que constituyen una red inalámbrica de sensores (véase la Figura 1.2), realiza una función especial y fundamental para que los fenómenos de estudio puedan ser monitorizados correctamente. A continuación se lista y explica cada uno de esos elementos:

- *Sensores*: toman del medio la información y la convierten en señales eléctricas. Existen de distinta naturaleza y tecnología.
- *Nodos*: toman los datos del sensor a través de sus puertas de datos, y envían la información a la pasarela o directamente a la estación base. Contienen un procesador que les permite realizar computaciones locales con la información captada por los sensores.
- *Pasarelas o gateways*: elementos para la interconexión entre la red de sensores y una red TCP/IP. Su función principal es la de recolectar los datos enviados por los nodos y transmitirlos hacia la estación base. Normalmente, las redes que interconectan suelen utilizar protocolos y arquitecturas diferentes, de modo que estos elementos actúan de traductor para convertir el formato de la información utilizado en la red de origen, al formato empleado en la red de destino.
- *Estaciones base*: recolector de datos basado en un ordenador común o sistema embebido. Los datos proceden normalmente de una pasarela, aunque pueden hacerlo también directamente desde los nodos. Es donde se realiza el mayor número de operaciones de cómputo del sistema, ya que no suele presentar restricciones críticas de tamaño y energía, al contrario que en los nodos. Además, pueden enviar órdenes de actuación tras llevar a cabo el análisis de los datos recibidos. Por ejemplo, en un sistema que controle el regadío de un campo de frutales, la estación central puede mandar activar el riego de los árboles tras detectar, según la información recibida, que éstos necesitan agua.
- *Red inalámbrica*: basada comúnmente en el estándar 802.15.4 *ZigBee* (ver la Figura 1.3). La conectividad inalámbrica de un importante número de aplicaciones tiene modestos requerimientos respecto a la transmisión de datos,

pero sin embargo demanda fiabilidad y seguridad en la comunicación, usando sistemas radio de bajo coste y escaso derroche energético. El estándar IEEE 802.15.4 y la tecnología inalámbrica ZigBee han sido diseñadas para satisfacer la necesidad del mercado de un estándar de bajo coste y de una tecnología de red inalámbrica flexible, la cual ofrezca un bajo consumo de la potencia, fiabilidad, interoperabilidad y seguridad para el control y monitorización de aplicaciones con tasas de datos bajas o moderadas [7].

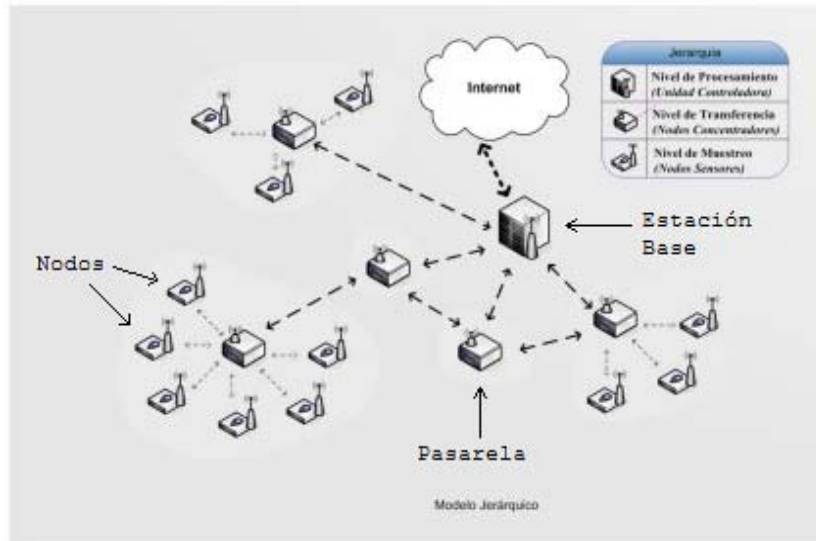


Figura 1.2 Elementos de una WSN

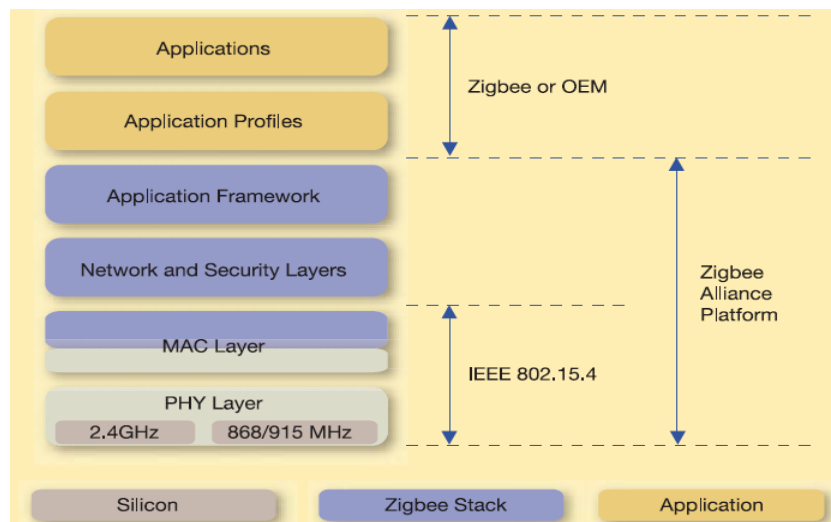


Figura 1.3 Modelo de trabajo de IEEE 802.15.4 y ZigBee

Un concepto muy importante de este tipo de redes inalámbricas es tener en cuenta que su diseño puede llegar a ser muy distinto en función del tipo de aplicación a

desempeñar. Sin embargo, todas las WSN creadas tienen las siguientes características comunes:

■ Gran Escala.

La cantidad de nodos que se despliega en una red puede crecer a lo largo de la vida de ésta, pudiendo llegar a contener miles de nodos. La red se va a componer de muchos sensores densamente desplegados en el lugar donde se produce el fenómeno y, por lo tanto, muy cerca de él.

■ Topología variable.

La posición en que se coloca cada uno de los nodos será de forma arbitraria o estratégica, y normalmente es desconocida por el resto de nodos. La localización no tiene porqué estar diseñada ni preestablecida, lo que va a permitir un despliegue aleatorio en terrenos inaccesibles u operaciones de alivio en desastres. Por otro lado, los algoritmos y protocolos de red deberán ser auto-organizativos.

■ Recursos limitados.

Los sensores de la WSN, debido a las exigencias de bajo consumo de potencia, poco coste y pequeño tamaño, disponen de recursos limitados, tales como procesadores de baja frecuencia o memorias de poca capacidad.

■ Cooperación de nodos sensores.

Realizan operaciones simples antes de transmitir los datos, lo que se denomina un procesamiento parcial o local. Esto permite ahorro en la computación a otros elementos de la red, así como una posible reducción del tráfico a través de ella.

■ Comunicación.

Los nodos sensores usan comunicación por difusión, y debido a que están densamente desplegados, los vecinos están muy cerca unos de otros y la comunicación multi-hop (salto de uno a otro múltiple) consigue un menor consumo de potencia que la comunicación single-hop (salto simple). Además, los niveles de transmisión de potencia se mantienen muy bajos y existen menos problemas de propagación en comunicaciones inalámbricas a largas distancias.

■ Funcionamiento autónomo.

Comúnmente, los nodos son capaces de desempeñar su función sin la necesidad de supervisión humana. Además, durante el tiempo en el que el nodo

permanece sin ser atendido porque no se puede acceder a él físicamente, puede ocurrir que se lleguen a agotar sus baterías o que su funcionamiento deje de ser el esperado.

Además, de forma muy general, las Redes de Sensores Inalámbricas se pueden dividir en dos grandes grupos según su arquitectura:

- Centralizada.

Los nodos de la red envían los paquetes con información a la pasarela o gateway más cercano, el cual se encarga de procesarlos y transmitirlos a la estación base. Esta configuración de la WSN puede propiciar que su tiempo de vida se acorte como resultado de un mayor consumo de energía debido al masivo número de mensajes que viajan por la red. Además, este aumento del tráfico ocasionaría problemas de saturación de red, al existir un cuello de botella en las pasarelas (véase la Figura 1.4).

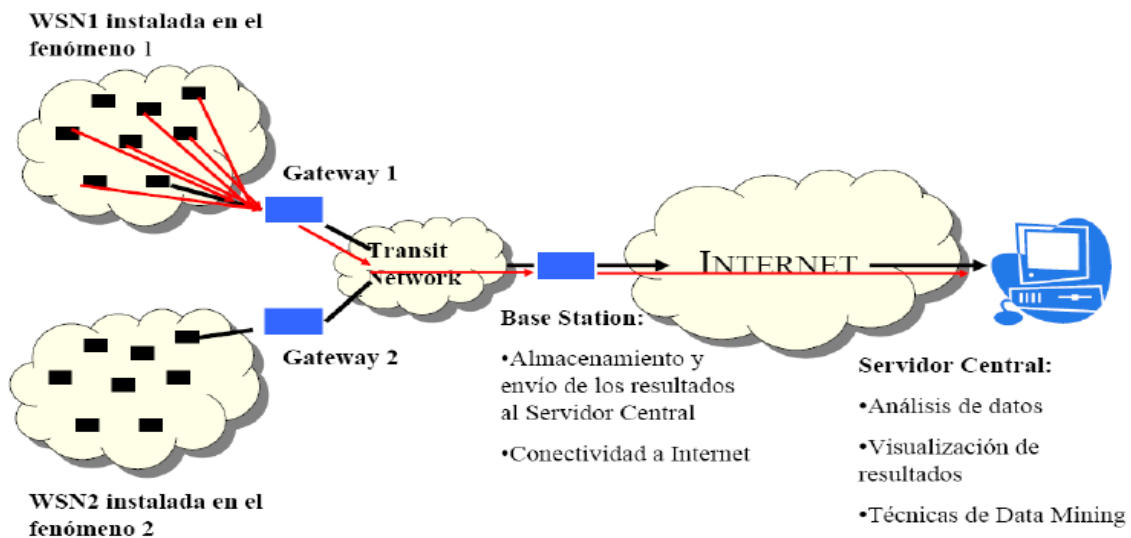


Figura 1.4 Arquitectura centralizada de una WSN.

- Distribuida.

En WSNs, normalmente se tiende a un tipo de arquitectura con computación distribuida. Los nodos sensores se van a comunicar con sus vecinos, y juntos van a

cooperar, ejecutando algoritmos distribuidos, para así obtener una única respuesta global, que un nodo especial, llamado *cluster head*, se encargará de transmitir a la estación base a través de la pasarela apropiada. Gracias a esta arquitectura se evitan los cuellos de botella en los gateways y existe una mayor eficiencia energética (véase la Figura 1.5).

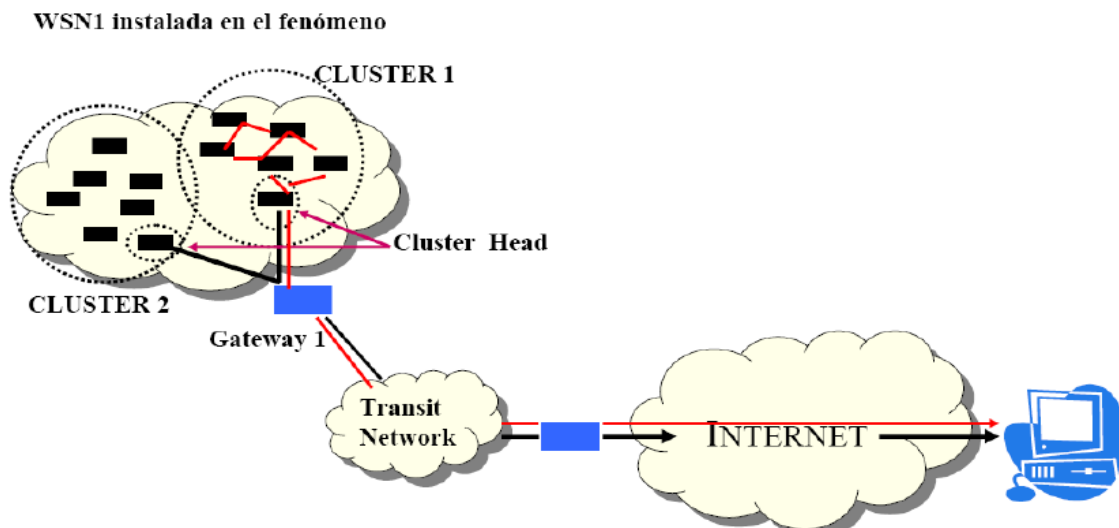


Figura 1.5 Arquitectura distribuida de una WSN.

Cuando una WSN es desplegada, su diseñador deberá seguir diversas pautas. Los requisitos necesarios o conjunto de áreas a considerar en el diseño son:

■ Topología.

La topología es generalmente cambiante para las Redes de Sensores Inalámbricas. Es decir, los nodos se despliegan de manera aleatoria. Una vez situados en el escenario no se requiere la intervención humana, haciendo la configuración y el mantenimiento completamente autónomos. En la organización de la topología, cada mote busca información completa de la red o parte de ella, con el fin de mantener las estructuras de información de la red actualizadas.

■ Enrutamiento.

La responsabilidad de un protocolo de enrutamiento es el intercambio de información, encontrando el camino más confiable para alcanzar el destino deseado teniendo en consideración la distancia, el requerimiento mínimo de energía y el tiempo de vida del enlace inalámbrico; búsqueda de la información en el caso de

que la conexión falle; y reparación de los enlaces caídos gastando el mínimo de potencia de procesamiento y ancho de banda.

- Energía.

Una cuestión capital en la operación de los nodos es la capacidad de energía. Los sensores tienen una vida intrínsecamente dependiente del tipo de batería que se utiliza (Ni-Cd, Ni-Mh, Li-Ion). Así mismo, el hardware diseñado para los nodos deberá tener un consumo óptimo de energía como requerimiento primordial. El microcontrolador, el sistema operativo y el software de aplicación deberán optimizar la conservación de energía, inclusive desde el punto de escoger las instrucciones y su codificación, para que las secciones de código más utilizadas empleen la menor potencia posible. Otros diseños optimizan el consumo de energía con la minimización de sistemas síncronos y/o el concepto de sistemas asíncronos globales y síncronos solamente en pequeñas porciones locales que lo necesiten.

- Sincronía.

Los sensores deberán ser capaces de sincronizarse unos con otros de manera completamente distribuida, para que la multiplexación en tiempo pueda ser impuesta y se ordene temporalmente la detección de eventos sin ninguna ambigüedad. Dado que los nodos de este tipo de redes inalámbricas operan de forma independiente, sus relojes podrán o no, estar sincronizados. Esto podría causar dificultades cuando se trate de integrar e interpretar información recogida en diversos nodos.

- Escalabilidad.

A medida que el número de sensores aumenta, la probabilidad de que existan fallos en la comunicación entre los nodos se incrementa de manera exponencial. Como consecuencia se requiere de un mejor control que mantenga a la red conectada a pesar del decremento en el ancho de banda utilizable. La escalabilidad y la certeza en la red son cuestiones que se contraponen, ya que al tener una red más densa, es más probable tener problemas en la comunicación.

- Calidad de servicio.

La calidad del servicio por enlace puede interpretarse en las WSNs, por flujo de información o por funcionamiento de nodo. En estos sistemas, tanto la red como el *host* pueden requerir situaciones que necesiten de una buena coordinación. Inconvenientes en dicha coordinación a nivel central y en un conjunto de determinados recursos puede desencadenar un grave problema en la red. Además,



la comunicación en tiempo real sobre una red de sensores deberá ser garantizada a pesar de existir un máximo retraso, un ancho de banda mínimo y otros parámetros involucrados en la calidad del servicio.

■ Tolerancia a fallos.

La red debe ser capaz de modificar algún aspecto de ella cuando se presente un determinado fallo. Éste deberá de ser identificado y resuelto en un tiempo promedio estipulado. Realizar este proceso con éxito es muy importante, primordialmente para aplicaciones militares y civiles donde existe un riesgo de pérdidas de vidas por el fallo de algún equipo electrónico.

■ Seguridad.

Las comunicaciones inalámbricas viajan por un medio fácilmente accesible a personas ajenas a la red de sensores. Esto implica un riesgo potencial para los datos recolectados y para el correcto funcionamiento de la red. Por consiguiente, se deben establecer mecanismos que permitan tanto proteger la información de estos intrusos, como protegerse de los datos que éstos puedan inyectar en la red.

■ Auto-organización.

Es una de las propiedades más importantes que se debe exhibir en una WSN. Entre las principales actividades que se efectúan mediante la autoorganización está la búsqueda de nodos adyacentes, la organización de la topología, así como la reorganización de ésta. Durante la fase de búsqueda, cada nodo de la red recolecta información acerca de los vecinos y la almacena en estructuras. Por ende, se demanda que periódicamente se transmita pequeños paquetes llamados *beacons*, que vigilan en el canal para detectar cualquier actividad del vecino. En la fase de reorganización se requiere información confiable de la topología, ya sea en base a la movilidad de los nodos, la cantidad de energía de éstos o la eficiencia en los enlaces, en caso de que se haya suscitado un cambio de último momento.

## 1.3 Aplicaciones

Existen diversos caminos de empleo de las redes inalámbricas de sensores, los cuales dependen de las funciones que éstas tengan que desempeñar. Los cuatro modos de uso principales son:

- Monitorización sincronizada. Una gran suma de nodos organizados y sincronizados miden y transmiten datos periódicamente. Propio de aplicaciones tales como las que controlan la agricultura o monitorizan algún microclima.
- Monitorización en base a eventos. Aplicación para detección de anomalías o ataques en entornos monitorizados continuamente por sensores. En estos casos, los nodos no están continuamente enviando datos, haciéndolo solamente en caso de que ocurra algún evento. Ejemplos: control de edificios inteligentes, detección de incendios, aplicaciones militares, etc.
- Monitorización para seguimientos. Aplicación para controlar objetos que están etiquetados con nodos sensores en una región determinada. La topología de la red es muy dinámica. Un ejemplo sería el rastreo de una comunidad de animales si se desea estudiar su comportamiento.
- Redes híbridas. Escenarios que contienen aspectos de las tres categorías anteriores.

Respecto al tipo de aplicaciones que las Redes de Sensores Inalámbricas pueden abarcar, existen de muy dispar naturaleza. Se va a intentar cubrir el mayor rango de ellas, clasificándolas en: militares, entorno, salud, hogar y otras áreas comerciales.

### 1.3.1 Aplicaciones militares

Las WSN pueden ser parte integral de sistemas militares C4ISRT que son los que llevan las órdenes, el control, comunicaciones, procesamiento, inteligencia, vigilancia, reconocimientos y objetivos militares.

El rápido y denso despliegue de las redes de sensores, su autoorganización y tolerancia a fallos las hace una buena solución para este tipo de aplicaciones. Ofrecen

una solución de bajo coste y fiable para éstas ya que la pérdida de un nodo no pone en riesgo el éxito de las operaciones. Ejemplos de aplicación en esta área son:

- Monitorización de fuerzas aliadas, equipamientos y munición.
- Reconocimiento del terreno y fuerzas enemigas.
- Adquisición de blancos.
- Valoración de daños.
- Reconocimiento de ataques nucleares, biológicos y químicos.

### 1.3.2 Aplicaciones medioambientales

En este ámbito tenemos aplicaciones como el seguimiento de aves, animales e insectos; monitorización de condiciones ambientales que afectan al ganado y las cosechas; irrigación; detección química o biológica; agricultura de precisión; detección de incendios; investigación de meteorología o geofísica; mapeado de la biocomplejidad del entorno; detección de inundaciones, etc. A continuación se destacan algunas de ellas:

- Detección de fuego en bosques. Se podrían desplegar millones de sensores de manera estratégica en el bosque que informaran del origen exacto de un incendio antes de que se haga incontrolable. Los sensores podrían tener métodos de obtención de energía como placas solares, ya que serían abandonados durante meses o años sin mantenimiento. Además, debido a la densidad de su despliegue, serían capaces de cooperar para realizar una recolección de datos cooperativa y evitar obstáculos en las transmisiones, como árboles y rocas que pueden bloquear la línea de visión entre sensores.
- Mapeado de la biocomplejidad del entorno medioambiental. Requiere enfoques sofisticados para integrar información en las escalas temporal y espacial. Avances tecnológicos en la adquisición de datos remota y automatizada han permitido una resolución espacial, espectral y temporal con un coste por unidad de área que se ha reducido en progresión geométrica. Además, la conexión de

estos sistemas a Internet permite a usuarios remotos controlar, monitorizar y observar la biocomplejidad del medio ambiente.

Aunque los satélites y sensores aéreos son útiles en la observación a gran escala de la biodiversidad, no tienen la precisión suficiente como para observarla a pequeña escala, que es la parte más importante de ésta en un ecosistema. Por ello, se necesita un despliegue de nodos sensores.

- Detección de inundaciones. Hay sensores de lluvia, nivel de agua y meteorológicos. Proporcionan información a una estación base centralizada. Hay proyectos de investigación que estudian enfoques distribuidos en interacción con los nodos sensores en campo para proporcionar consultas en momentos fijos o en espacios temporales largos. Un ejemplo donde una WSN fue desplegada para la detección de inundaciones, es el sistema ALERT de EEUU.
- Agricultura de precisión. Beneficios obtenidos de monitorizar niveles de pesticidas en agua potable, niveles de erosión del suelo y niveles de polución del aire en tiempo real.

### 1.3.3 Aplicaciones sanitarias

Chips como el mostrado en la Figura 1.6, hasta ahora confinados en las máquinas, comienzan a adaptarse al cuerpo humano. Entre la ciencia de ficción y la de realidad, la idea es crear Redes de Sensores Inalámbricas que permitan un control del estado de salud de las personas y sirvan para prevenir problemas y enfermedades. Con esta ayuda tecnológica, el hombre podría anticiparse a muchas patologías como los ataques al corazón. Algunas de las aplicaciones en estudio son: un chip con historial médico, circuitos de control del corazón, el ojo y la mano biónica, tratamiento del cáncer de hígado, lectura del pensamiento, la neurona artificial y un antidepresivo.



**Figura 1.6** Chip para el estudio del cuerpo humano

### 1.3.4 Aplicaciones de hogar: domótica

En este ámbito los nodos sensores pueden ser introducidos en aparatos domésticos como aspiradoras, microondas, frigorífico, etc. Esto permite que sean manejados remotamente por los usuarios finales mediante una comunicación que se realizaría vía satélite o Internet.

A través de las redes de sensores pueden crearse hogares inteligentes donde los nodos se integran en muebles y electrodomésticos. En una misma habitación, estos nodos se comunican entre ellos y con el servidor de la habitación, que a su vez se comunican con otros servidores de otras habitaciones existiendo así una completa conectividad en toda la casa. Todos ellos se integran y organizan con los dispositivos integrados existentes para auto-organizarse, autorregularse y auto-adaptarse basándose en modelos de control.

De esta forma se crea lo que llamamos un entorno inteligente, cuyo diseño puede tener dos enfoques. Uno desde el punto de vista humano, donde el entorno se adapta a las necesidades del usuario final; y otro desde el punto de vista tecnológico, en el que se desarrollan nuevas tecnologías hardware, nuevas soluciones de redes y nuevos servicios middleware.

### 1.3.5 Otras aplicaciones comerciales

Actualmente, existen otras muchas aplicaciones comerciales de las Redes de Sensores Inalámbricas. Se van a describir a continuación tres más:

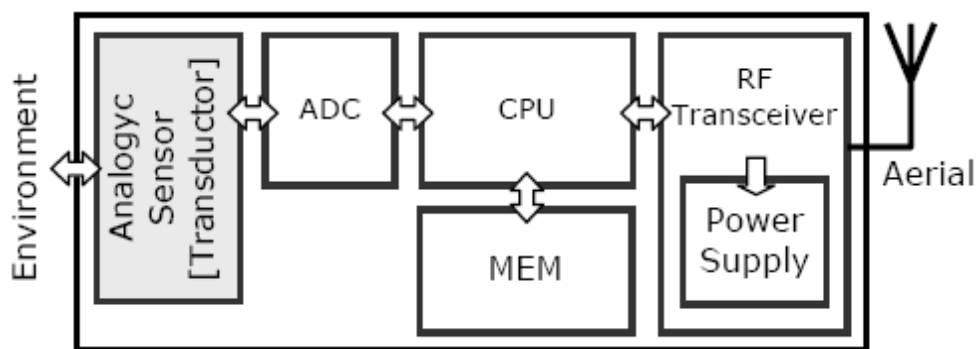
- Control ambiental en edificios de oficinas. Normalmente la calefacción o el aire acondicionado se controlan desde una central, por lo que la temperatura dentro de cada habitación puede variar unos pocos grados debido a que la distribución de aire no está uniformemente distribuida y el control es central. Así pues, se podría desplegar una WSN para controlar el flujo de aire y la temperatura en diferentes partes de la habitación.
- Museos interactivos. Serviría para que los niños interactúen con los objetos y experimentos, y reaccionen con el contacto o el habla. También permitiría el aviso y localización de personas en el recinto (Ejemplo: exploratorium en

museo de San Francisco).

- Gestión de inventarios. En un almacén, cada artículo llevaría pegado un sensor, de modo que se podría conocer en todo momento su localización y número de artículos existentes por categoría. Para dar de alta nuevos artículos se les pegaría el sensor y se llevarían al almacén.

## 1.4 Principales motes comerciales

Un nodo sensor es un elemento computacional con capacidad de procesamiento, de memoria, con interfaz de comunicación y que está formado por un conjunto de microsistemas sensores. Así pues, tal y como se puede observar en la Figura 1.7, el hardware básico de un dispositivo sensor se constituye de un transceptor, el cual permite la comunicación (envío y recepción de mensajes) entre el nodo y el resto de entidades de la red; un procesador, que controla la actividad del nodo; la memoria, para el almacenamiento de datos; la fuente de alimentación, la cual suele estar constituida por baterías aunque puede estar ayudado de un generador (ejemplo: placas solares); y los sensores, que son los responsables de captar los fenómenos exteriores y convertir las señales de analógico a digital, y así la información pueda ser tratada por el procesador.



**Figura 1.7** Estructura hardware básica de un sensor inalámbrico inteligente

Estos sensores nodos o motes son pequeños dispositivos inalámbricos de procesamiento basados en tecnología MEMS, que detectan valores físicos. Ellos están formados a su vez, como bien se ha indicado anteriormente, por unidades sensoras que pueden ser de naturaleza muy variada (sensores acústicos, sísmicos, infrarrojos, mecánicos, de calor, temperatura, radiación, etc.). La Figura 1.8 muestra algunos tipos de unidades

sensores que pueden presentarse en un mote, tal como un micrófono (Figura 1.8-a), que puede ser utilizado para captar señales acústicas del medio ambiente; un fotómetro (Figura 1.8-b), que es capaz de medir la cantidad de fotones recibidos; un diodo detector de luz (Figura 1.8-c), un acelerómetro (Figura 1.8-d), el cual mide la aceleración de un objeto; un sensor de humedad (Figura 1.8-e); un sensor de radiación (Figura 1.8-f); un sensor de luz ultravioleta (Figura 1.8-g); o un sensor de fuerza motriz (Figura 1.8-h).



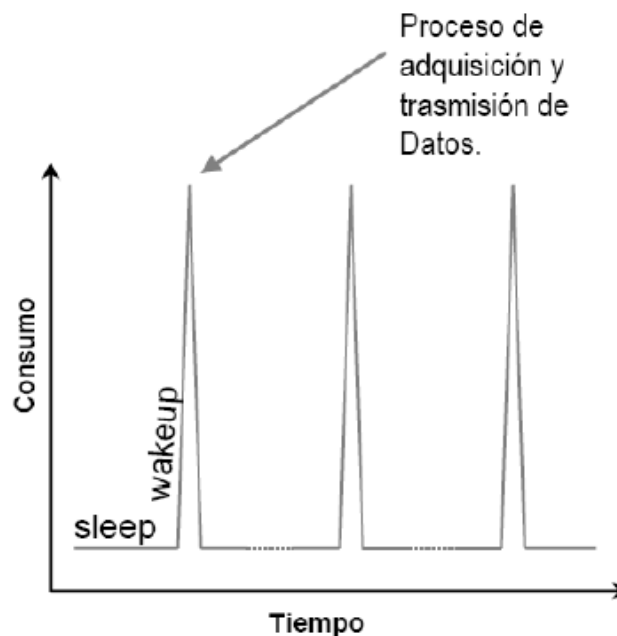
**Figura 1.8.** Ejemplos de unidades sensoras

Hoy en día, la tendencia es producir sensores a gran escala, a precios bajos y con la mejor capacidad de cómputo y de reducción de tamaño que sea posible. Es elemental resaltar que los nodos tienen fuertes restricciones en cuanto a la capacidad de memoria, de procesamiento y de energía principalmente, siendo deseable poseer microprocesadores de bajo consumo. Esto debe estar presente en el sistema operacional, siendo conscientes que la aplicación debe realizar un control exhaustivo del gasto de energía. Por ello, se han llevado a cabo ciertas estrategias hardware y software con el objetivo de obtener el mayor ahorro energético, de modo que el tiempo de vida del mote en la red con capacidad funcional sea el máximo posible.

Acorde a su modo de funcionamiento el nodo sensor puede permanecer en tres estados diferentes:

1. Sleep: es el estado en el que el mote está durmiendo o inactivo. Se pretende que la mayor parte del tiempo esté en este estado y por tanto su consumo sea el mínimo.
2. Wakeup: es el estado de cambio en el que el nodo se despierta y pasa a estado activo. Dicha transición se produce cuando el sensor recibe algún cambio, estímulo o interrupción programada dentro de sus funciones de detección y análisis. Uno de los objetivos es minimizar este tiempo para pasar rápidamente al estado de trabajo.
3. Active: es el estado activo del mote, donde está realizando el trabajo de adquisición, procesado y transmisión de datos. Por supuesto, este tiempo debe ser mínimo para volver cuanto antes al estado sleep, ya que su consumo es el mayor de los tres descritos.

En la Figura 1.9 podemos ver gráficamente los estados de un nodo sensor y su consumo para verificar lo anteriormente explicado.



**Figura 1.9** Consumo de un nodo sensor según su estado.

En las siguientes secciones se describen dos de los tipos de motes más populares del mercado actualmente: los Telos y Micas. Pero previamente serán descritos los nodos

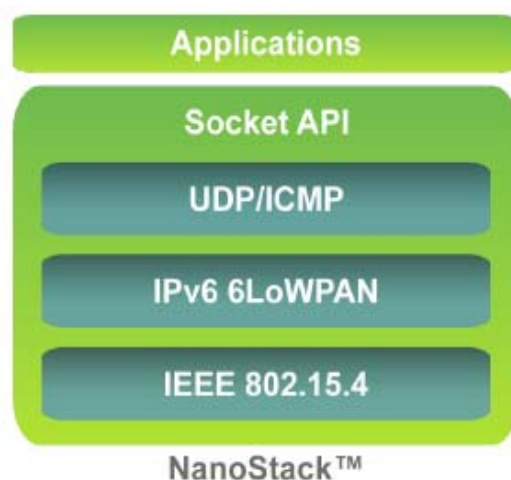


de las series de la compañía *Sensinode* [8], con los cuales se ha experimentado en el presente proyecto. Por tanto, a continuación se definen las características más importantes de estas tres familias de motes, y se realiza un análisis comparativo entre ellas.

### 1.4.1 Sensinode Series

Los productos para Wireless Sensor Networks de la compañía finlandesa *Sensinode* están basados en la tecnología pionera 6LoWPAN, la cual está también basada en IP. El uso de los estándares IEEE 802.15.4, IP y ZigBee, les permite una interoperabilidad global y un tiempo de vida elevado. Además, sus nodos sensores permiten realizar una amplia gama de distintas medidas de los fenómenos del medio, así como llevar a cabo el seguimiento de objetos, gente y áreas en tiempo real [9].

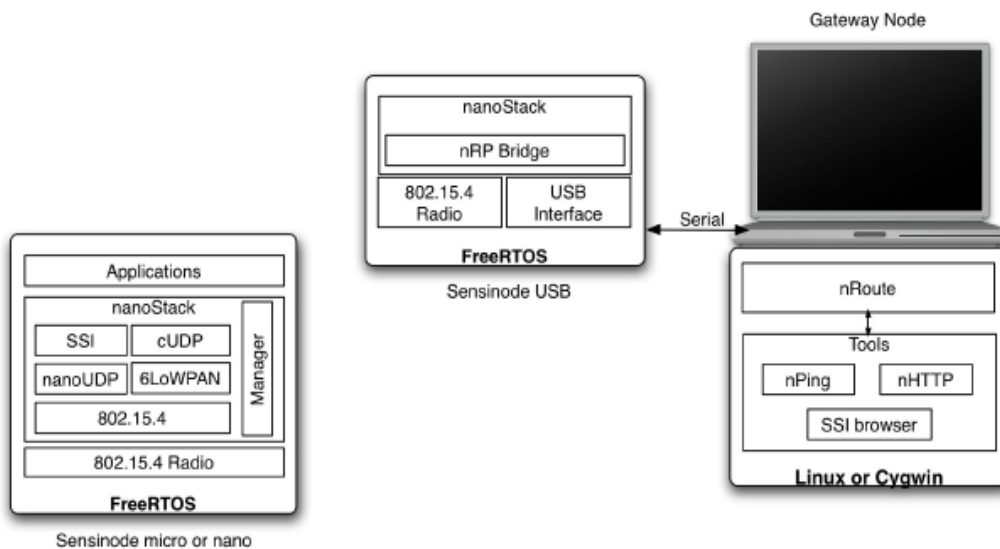
La característica principal de los nodos sensores producidos por la firma *Sensinode* es el uso de la pila de protocolos NanoStack [10], la cual se muestra en la Figura 1.10. Se trata del conjunto de protocolos, entre los que destaca 6LoWPAN, para la adquisición de datos inalámbrica y el control usado en dispositivos de baja potencia. NanoStack contiene completamente la implementación IEEE 802.15.4 e incluye además 6LoWPAN Ipv6, UDP, ICMP, IEEE 802.15.4 MAC y el protocolo sensor SSI. Además, protocolos cliente pueden ser añadidos a NanoStack como elementos de protocolo. Por otro lado, NanoStack es construido sobre el sistema operativo de código abierto llamado FreeRTOS. Aunque, por ejemplo, los productos de las Micro series de Sensinode también soportan TinyOS.



**Figura 1.10** Arquitectura de NanoStack

Cabe destacar también que NanoStack provee acceso a los datos intercambiados mediante comunicaciones a través de un fácil uso del interfaz Socket. Actualmente, este interfaz es ampliamente utilizado en las comunicaciones entre computadores (especialmente en los sistemas basados en POSIX), siendo también una herramienta muy útil para sistemas empotrados como los nodos sensores. La API de NanoStack es muy similar a la de POSIX y añade características para el manejo de la memoria en orden de conseguir flexibilizar las operaciones con buffers.

Esta pila de protocolos soporta tanto el entorno de desarrollo Linux como Windows (Cygwin) usando herramientas de código abierto (gcc, make, sdcc, etc.). En la Figura 1.11 podemos observar los componentes internos de la arquitectura NanoStack.

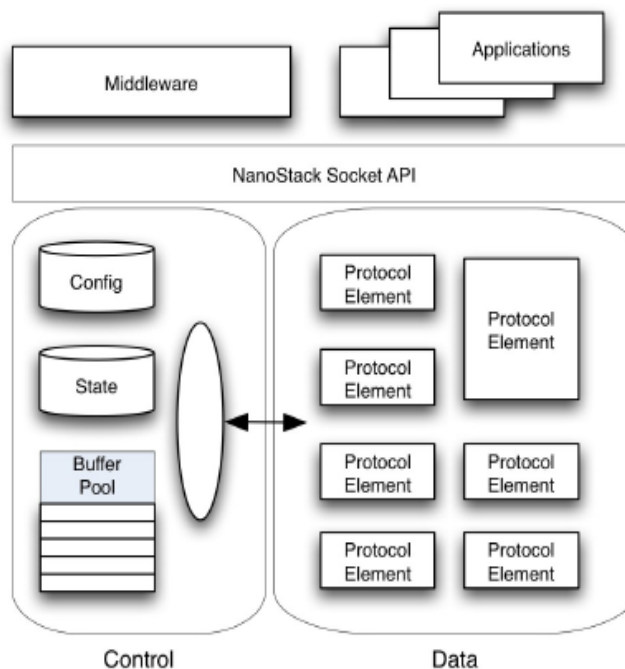


**Figura 1.11** Componentes internos de la arquitectura NanoStack.

Tal y como se nombro anteriormente, NanoStack es construido sobre un estable y portable sistema operativo en tiempo real llamado FreeRTOS. Éste provee un microkernel con un programador, un microcontrolador, memoria, colas, y semáforos, dentro de un sistema funcionalmente temporal. Además, es portable a una gran variedad de arquitecturas y compiladores [10].

NanoStack es ejecutada como una simple tarea en el entorno FreeRTOS. Esto permite reducir el uso de la RAM y proveer un control de flujo efectivo: los módulos de los protocolos son siempre ejecutados secuencialmente. Además, el uso de la pila esta simplificado también porque los módulos de los protocolos no emplean llamadas de funciones directas entre ellas. Por otro lado, los buffers se distribuyen a lo largo de una

única cola, lo cual asegura que la aplicación de usuario no está bloqueada durante la operación de la pila de protocolos (véase la Figura 1.12).



**Figura 1.12** Detalle de los componentes internos de NanoStack

A continuación vamos a describir brevemente algunos de los protocolos de la arquitectura NanoStack:

- IEEE 802.15.4.  
El estándar IEEE 802.15.4 [11] especifica un interfaz inalámbrico para la construcción de distintas aplicaciones wireless. Es muy flexible, permitiendo su aplicación desde redes ad-hoc hasta infraestructuras basadas en topologías de árbol. Teóricamente hasta un máximo de 64 mil dispositivos pueden estar interconectados en una única topología de esta especificación. La capa física de 2.4 GHz transmite datos a una tasa de 250 kbps y emplea 15 canales. Además, el estándar IEEE 802.15.4 es usado tanto por ZigBee como por 6LoWPAN IP.
- 6LoWPAN.  
6LoWPAN [12] es una especificación IETF que permite el uso de la compresión de los estándares IPv6 y UDP sobre redes IEEE 802.15.4. Normalmente las

cabeceras de los protocolos Ipv6 y UDP ocupan una gran cantidad de espacio, de forma que 6LoWPAN las comprime, siendo esto ideal para su uso en WSNs.

- NanoMesh.

Es una técnica implementada por *Sensinode* para que en una red 6LoWPAN se puedan cubrir múltiples saltos. NanoMesh usa la característica *mesh-header* de 6LoWPAN, implementando un algoritmo salto a salto, el cual hace uso de la tabla de encaminamiento que proporciona NanoStack. Además, esta técnica soporta auto-recuperación ante fallos y descubrimiento de gateways. Para redes de tamaño limitado (2 ó 3 saltos) que cambian rápidamente, NanoMesh es una excelente técnica.

- Protocolo SSI.

El protocolo SSI (Simple Sensor Interface) permite el descubrimiento y acceso a los sensores de una WSN.

- nRP.

El protocolo nRoute (nRP), es usado para la comunicación entre un PC host y un dispositivo serie para permitir el acceso del PC a la red de sensores local con el fin de poder monitorizarla, realizar diagnósticos o recolectar datos.

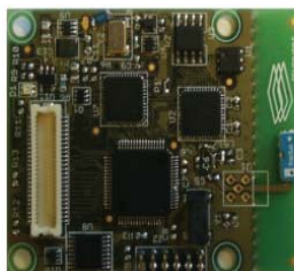
Los productos de la firma Sensinode se dividen en dos grandes conjuntos: las Nano series y las Micro series [9]. Ambos comparten las características de arquitectura descritas a lo largo de esta sección.

- Nano series: están formados por routers, nodos y módulos OEM (Original Equipment Manufacturer) para llevar a cabo el despliegue de sistemas (ver la Figura 1.13). Se utiliza una tecnología que optimiza la reducción del tamaño y coste de los dispositivos. Las Nano series están formadas por los productos: NanoModule, NanoSensor y NanoRouter.



**Figura 1.13** NanoSensor N711 de las Nano series de *Sensinode*

- Micro series: uno de los dispositivos pertenecientes a esta familia y mostrado en la Figura 1.14, el Micro.2420 U100, será testeado mediante experimentación en el presente proyecto. Los productos de esta serie son perfectos para su uso en I+D de Redes de Sensores Inalámbricas. Además, permiten máxima flexibilidad y reusabilidad. Como característica física más importante de su hardware indicar que los Micro series productos son bastante sensibles a las descargas electrostáticas (ESD), y por tanto es primordial que los dispositivos sepan manejar los posibles errores causados por ellas. Por último, en la Tabla 1.1 se listan todos los productos disponibles de las Micro series de Sensinode [13]:



**Figura 1.14** Sensor Micro.2420 U100 de las Micro series de *Sensinode*

PN#	Name	Notes
U100	Micro.2420	Also called Micro.4
U100R2	Micro.2420 version 2	
U510	Micro.compass	
U510R2	Micro.compass version 2	Enhanced power down modes
U600	Micro.USB	
U600R2	Micro.USB version 2	
U700	Micro.proto	
U500	Micro.data	
D100	Debug	Limited support for Nano-series programming
D100R2	Debug version 2	Full support for Nano-series programming
A500	Sensor IO board	
A100	USB programmer	special order only
K100	Sensinode Devkit	Sensinode Micro-series development kit 1
K110	Sensinode Devkit V2	Sensinode Micro-series development kit 2

**Tabla 1.1** Productos de la Micro series de Sensinode

## 1.4.2 Telos

Los modelos de mote TPR2400 y TPR2420 (“TelosB”) son plataformas de código abierto desarrolladas y publicadas por la Universidad de California, Berkeley (“UC Berkeley”). Cada uno de estos modelos contiene un procesador integrado (microcontrolador TI MSP430) que incluye un interfaz USB para su programación, 2.4 GHz radio, antena PCB y diferentes unidades sensoras. El modelo TPR2420 (ver Figura 1.15) es idéntico al TPR2400, a excepción de que el primero contiene un conjunto de unidades sensoras del entorno preinstalado.



**Figura 1.15** Modelo Telos TPR2420

Los TelosB poseen unas prestaciones que les permiten ser las herramientas ideales para trabajar en laboratorios. Sus principales aplicaciones son:

- Empleo en el desarrollo de aplicaciones de investigación de baja potencia.
- Utilización para la experimentación en Wireless Sensor Networks.

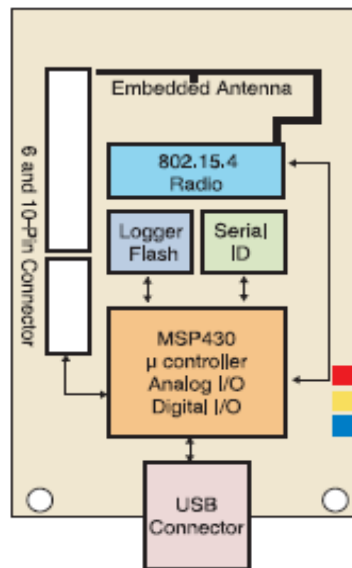
Esta plataforma consigue un bajo consumo de potencia permitiendo una larga vida a las baterías además de tener un tiempo mínimo en el estado de wakeup, otro de los objetivos dentro de las estrategias de bajo consumo. Además, los TelosB se alimentan de 2 baterías AA, aunque si son conectados mediante el puerto USB para programación o comunicación, la alimentación la proporciona el ordenador.

También se tiene la capacidad de añadir dispositivos adicionales. Los dos conectores de expansión de los que dispone pueden ser configurados para controlar sensores analógicos, periféricos digitales y displays LCD. En la Figura 1.16 se puede ver el diagrama de bloques del mote TPR2420.

Por otro lado, sus principales características son [14]:

- Transmisión RF de acuerdo con la norma IEEE 802.15.4/ZigBee.
- Banda de frecuencias desde 2.4 a 2.4835 GHz.
- Velocidad de transferencia de datos de 250 kbps.
- Antena integrada.
- Micro-controlador MSP430 a 8MHz con 10kB de RAM.
- Bajo consumo.
- Memoria flash externa de 1 Mb para almacenamiento de datos.
- Programación y toma de datos vía USB.
- Conjunto opcional de sensores de luz, temperatura y humedad (TPR2420).

- Soporta TinyOS 1.1.10 o versiones más actualizadas.



**Figura 1.16** Diagrama de bloques del modelo TPR2420

Con todas estas características, el mote TelosB no sólo proporciona más facilidad para programación, más flexibilidad y más prestaciones, sino que es el que menos consumo de potencia ofrece permitiendo alargar la vida de los nodos considerablemente y siendo ésta su principal baza frente a los otros dispositivos.

### 1.4.3 Micas

En los siguientes apartados se van a explicar las características más destacadas de los principales motes de esta familia: MicaZ, Mica2 y Mica2Dot. Todos ellos se programan utilizando la *Placa de desarrollo* MIB510 [15], mostrada en la Figura 1.17. Esta placa actúa como interfaz entre el PC y los motes a través del puerto serie, permitiendo la programación de los dispositivos acoplados. Esta placa tiene un procesador (ISP) ATmega16L mediante el cual se programan los motes. El código se descarga al ISP a través del puerto serie RS-232, y es el ISP el que programa el código en el mote.

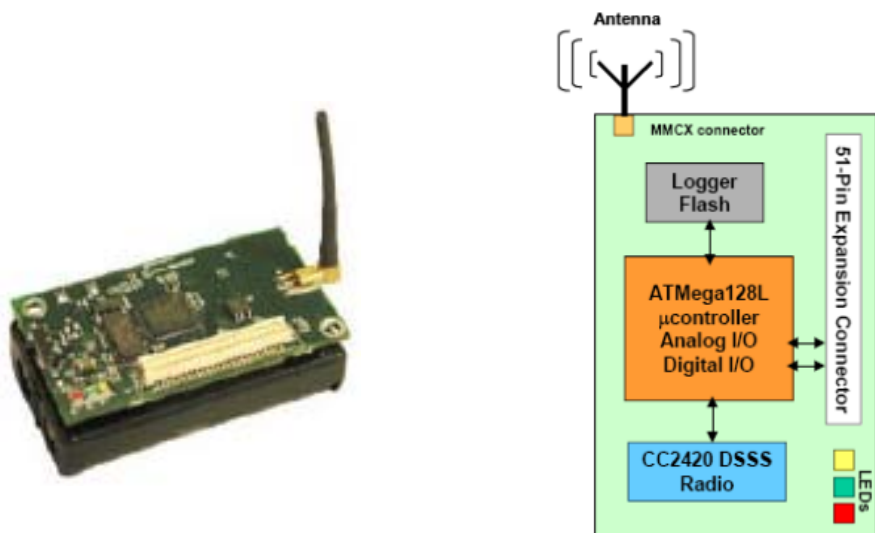




Figura 1.17 Placa de desarrollo MIB510

■ **MicaZ.**

Son unos de los motes de las últimas generaciones producidos por Crossbow Technology. En la Figura 1.18 se muestra el aspecto del dispositivo y su diagrama de bloques de este dispositivo. El mote MicaZ [16] está embebido con un transceptor ZigBee de radio frecuencia que trabaja en la banda de 2400 MHz a 2483.5 MHz y utiliza modulación digital de espectro esparcido de secuencia directa (DSSS), ofrece una velocidad de 250 Kbps y seguridad por hardware (AES-128). El conector de expansión soporta interfaces de entradas analógicas y entradas/salidas (I/O) digitales. Además usa el Chipcon CC2420, bajo la norma del estándar IEEE 802.15.4 y un microcontrolador ATmega128L para la programación de los motes.



a) Mote

b) Diagrama de bloques

Figura 1.18 MicaZ

■ **Mica2.**

Los motes Mica2 (ver la Figura 1.19) son los módulos de tercera generación de motes que se usan para Redes de Sensores Inalámbricas de baja potencia [17]. Mejoran las características del Mica original:

Diseñado específicamente para redes de sensores integradas.

- Distintas frecuencias de transmisión con amplio rango.
- Más de un año de batería mediante los modos sleep.
- Permiten reprogramación inalámbrica a distancia.
- Amplia variedad de placas de sensores compatibles: luz, temperatura, presión, aceleración, acústica, etc.



**Figura 1.19** Mica2

Las distintas aplicaciones en las que se utilizan estos motes son, principalmente, las WSN, la seguridad y vigilancia, la monitorización ambiental o las redes inalámbricas de gran escala.

Los Mica2 se dividen en tres modelos dependiendo de su frecuencia de uso: MPR400 (915MHz), MPR410 (433MHz), y MPR420 (315MHz). Estos motes usan el Chipcon CC1000, con radio modulada en FSK. Todos los modelos usan un potente microcontrolador ATmega128L y una radio de frecuencia sintonizable en un rango amplio. En la Figura 1.20 se muestra el diagrama de bloques del mote.

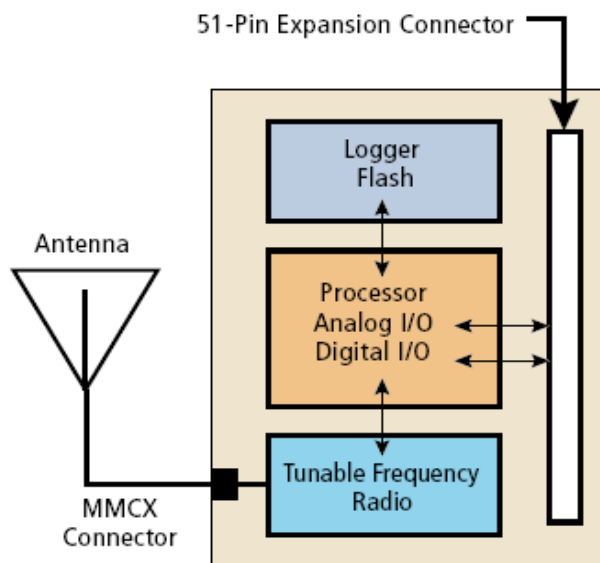


Figura 1.20 Diagrama de bloques del modelo MPR400 de Mica2

■ **Mica2Dot.**

Los Mica2Dot son un tipo de motes diseñados especialmente para aplicaciones donde el tamaño físico es fundamental. Al igual que los Mica2 hay tres modelos dependiendo de su frecuencia: MPR500 (915MHz), MPR510 (433MHz), y MPR520 (315MHz). El resto de características son similares a las de los Mica2, lo más relevante es la forma física y el reducido tamaño que poseen, tal y como podemos ver en la Figura 1.21.

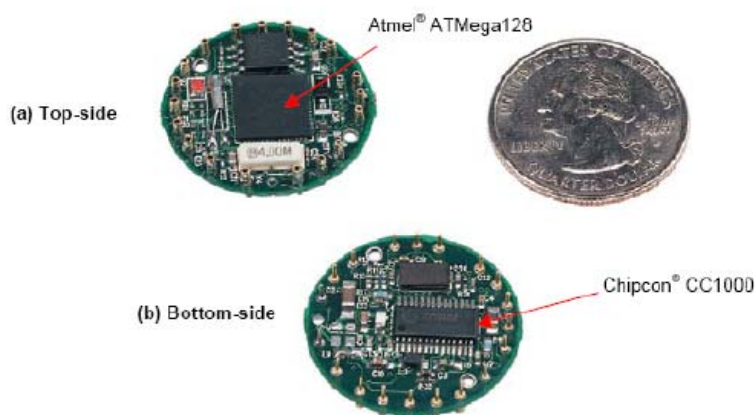
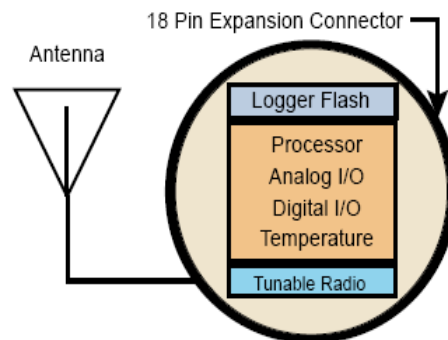


Figura 1.21 Mica2Dot

Al igual que el resto de motes Mica explicados anteriormente, está basado en el microcontrolador Atmel ATmega128L de baja potencia y opera con TinyOS desde su memoria flash interna. Tiene 18 pines de expansión para conectar 6 entradas

analógicas, I/O digital y una interfaz de comunicaciones serie o UART. En la Figura 1.22 puede apreciarse el diagrama de bloques del modelo MPR500.










**Figura 1.22** Diagrama de bloques del modelo MPR500 de Mica2Dot

Sus características más importantes son [18]:

- Transmisión multi-canal 868 / 916 MHz, 433 MHz o 315 MHz de rango extendido.
- Permite programación remota inalámbrica.
- Compatible con el mote Mica2.
- Dispone de un sensor de temperatura, monitor de la batería y LED.

#### 1.4.4 Comparación entre tecnologías

Existen evidentes diferencias entre las tres familias de motes descritas en las subsecciones anteriores. Desde el punto de vista de la arquitectura y hardware los dispositivos que presentan más similitudes entre ellos son los Mica y Telos. Estos últimos y los MicaZ pueden trabajar en la banda de 2.4 GHz. Además los dispositivos de los dos grupos emplean Zigbee para la comunicación RF y soportan TinyOS. Sin embargo, los Mica se programan por el interfaz serie RS-232, mientras que los Telosb lo hacen por el USB. También poseen microcontroladores distintos, ya que todos los Mica utilizan el Atmega128L, y los Telosb el MSP430. En la Tabla 1.2 se puede observar la evolución de los distintos prototipos y un resumen comparativo de las características de los sensores de estas dos familias.

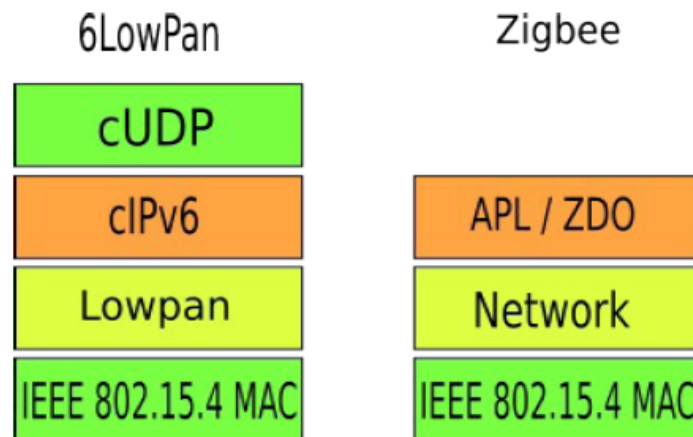
Mote Type Year	WeC 1998	René 1999	René 2 2000	Dot 2000	Mica 2001	Mica2Dot 2002	Mica 2 2002	Telos 2004
								
Microcontroller	AT90LS8535		ATmega163		ATmega128			TI MSP430
Program memory (KB)	8		16		128			60
RAM (KB)	0.5		1		4			2
Active Power (mW)	15		15		8	33		3
Sleep Power ( $\mu$ W)	45		45		75	75		6
Wakeup Time ( $\mu$ s)	1000		36		180	180		6
Nonvolatile storage	24LC256				AT45DB041B			ST M24M01S
Connection type	I <sup>2</sup> C				SPI			I <sup>2</sup> C
Size (KB)	32				512			128
Communication	TR1000				TR1000	CC1000		CC2420
Data rate (kbps)	10				40	38.4		250
Modulation type	OOK				ASK	FSK		O-QPSK
Receive Power (mW)	9				12	29		38
Transmit Power at 0dBm (mW)	36				36	42		35
Power Consumption	2.7		2.7		2.7			1.8
Total Active Power (mW)	24				27	44	89	41
Programming and Sensor Interface								
Expansion	none	51-pin	51-pin	none	51-pin	19-pin	51-pin	10-pin
Communication	IEEE 1284 (programming) and RS232 (requires additional hardware)							USB
Integrated Sensors	no	no	no	yes	no	no	no	yes

**Tabla 1.2** Resumen comparativo de distintos motes

Tal y como se puede apreciar en la tabla, el microcontrolador MSP430 de los dispositivos Telos consume una cantidad significativa de energía menos que el de los motes Mica. La velocidad de transferencia es bastante más elevada también en los Telos (250 kbps), presentando un tipo de modulación distinta al resto de nodos Mica. Además, el hecho de que los Telos lleven a cabo la programación vía USB y que las unidades sensoras se encuentren integradas ya en el dispositivo, le otorga más facilidad de uso respecto a los Mica.

Por otro lado se encuentran los productos de las series de *Sensinode*. Estos se diferencian principalmente de los otros dos grupos de motes en la arquitectura de protocolos de comunicación, ya que las Micro y Nano series emplean la pila NanoStack. Su característica más importante es la presencia en esa pila del protocolo 6LoWPAN, el cual realiza la compresión de las cabeceras de UDP e IPv6, propiciando significativas ventajas energéticas, ya que el hecho de enviar paquetes de menor tamaño causa que los nodos de la red puedan estar menos tiempo transmitiendo y por tanto más tiempo sin realizar operaciones, lo que supone ahorro de batería. En la

Figura 1.23 se puede apreciar una comparación entre los protocolos 6LoWPAN y Zigbee, el cual es el utilizado por los motes Mica y Telos.



**Figura 1.23** Protocolos Zigbee y 6LoWPAN comparados

Normalmente NanoStack es ejecutado sobre FreeRTOS, aunque puede ser también empleado con TinyOS. El hecho de poder funcionar sobre FreeRTOS, le proporciona a los sensores de *Sensinode* un mejor aprovechamiento de recursos, tales como la memoria, ya que el programa es ejecutado internamente como una tarea simple y secuencial, siendo esto otra importante ventaja. Otra de ellas sería el empleo del interfaz USB para llevar a cabo la programación. Además, los sensores de la serie Micro (con los cual experimentamos en este proyecto) utilizan el microprocesador MSP430 que como vimos en la Tabla 1.2 presenta muy buenas prestaciones respecto al ahorro energético.

Así pues, los motes de *Sensinode* presentan argumentos de peso que nos permiten considerar que pueden poseer importantes prestaciones respecto al ahorro energético, la cual es una característica crucial para cualquier red inalámbrica de sensores. Otros dispositivos que 'cuidan' de forma excelente este importante factor son los motes Telosb, mientras que los Mica presentan peores resultados.



# Capítulo 2

## Metodología para la caracterización de nodos en WSNs

---

### 2.1 Introducción

El término *metodología* hace referencia al conjunto de métodos que se siguen en una investigación científica. De este modo, dentro del contexto de las Redes de Sensores Inalámbricas, Zhao y Govidan [2] lo definieron como el conjunto de técnicas sistemáticas que permiten estudiar el comportamiento de las redes de sensores en escenarios diferentes. Ese es el objetivo fundamental de este trabajo: definir una metodología que describa minuciosamente todo el proceso que un usuario debe llevar a cabo con el fin de investigar y averiguar cuál es el comportamiento de cualquier tipo de nodo sensor en diferentes escenarios.

Tras la ejecución de la metodología, se dispondrá de un conjunto de datos, que tras su análisis le permitirán al diseñador conocer la respuesta real en el escenario seleccionado para los nodos sensores con los que se ha experimentado. Esta metodología se puede ejecutar sobre cualquier juego de sensores del mercado, sirviendo esto como un test de los dispositivos antes de que ellos sean desplegados en la red de sensores final.

El comportamiento real de un determinado conjunto de sensores dependerá de la respuesta de éstos al estudio, mediante experimentación, de ciertos parámetros o características, tal como la probabilidad de pérdida de paquetes. Por tanto, su comportamiento, que es lo que se quiere averiguar, es definido por esos parámetros, y por consiguiente, la metodología de experimentación debe definirse con el fin de que el mayor número posible de esos parámetros sean estudiados.

Posiblemente al diseñador de la WSN no le interese testear una gran cantidad de estos parámetros, sino que quizás un pequeño conjunto de ellos le es suficiente, debido a que los requerimientos de la red sólo le obligan a realizar un estudio crítico de ciertas características, como por ejemplo la vida de las baterías de los sensores o la orientación de éstos. Así pues, las necesidades de cada red de sensores pueden



ser muy dispares, dándose en este proyecto una guía general de cómo se pueden testear las características de un nodo.

Los parámetros del nodo sensor testeados por la presente metodología son cinco: la tasa de recepción de paquetes, la fuerza de la señal recibida, el indicador de calidad del enlace, el ángulo de orientación y la vida de la batería. Todos ellos serán descritos en subsecciones posteriores. La tasa de colisiones o la respuesta del sensor con la temperatura ambiente son algunas de las características que no han sido incluidas en esta experimentación.

Estos parámetros o características de los nodos de una WSN presentan diferentes respuestas según ciertos fenómenos o variables. La influencia de éstos es vital sobre el comportamiento de los motes y por tanto sobre el desempeño final de la red. Por ello, es crucial identificarlos y estudiarlos correctamente. La presente metodología analizará una gran cantidad de ellos, tales como la localización de la red o la potencia de transmisión de los sensores.

Por otro lado, la metodología de testeo está dividida en tres experimentos, siendo investigados mediante ellos los cinco parámetros citados anteriormente. Se ha dividido en tres pruebas distintas debido a que para cada una de ellas es diferente tanto la configuración del escenario, como la disposición de los elementos y los parámetros de los sensores testeados.

Así pues, antes de describir los tres experimentos principales de la metodología, en el siguiente subapartado se explican las variables y fenómenos que provocan los cambios en el comportamiento de los sensores. Seguido, para cada experimento se definirán los parámetros que estudia y el procedimiento mecánico completo de cómo se lleva a cabo el testeo de éstos.

## **2.2 Variables experimentales**

Cualquier sistema de comunicación es afectado por numerosos fenómenos. En el ámbito de las Redes de Sensores Inalámbricas, la variación de cada uno de estos fenómenos puede provocar un cambio drástico en el comportamiento de los elementos que forman el sistema. Por ello, es muy importante identificarlos y definirlos de forma precisa en la metodología de experimentación, con el fin de realizar un amplio estudio

práctico de ellos cuando se lleven a cabo los tests y averiguar cual es su repercusión sobre el comportamiento de los elementos de la WSN.

Estos fenómenos o variables pueden ser divididos en dos grupos principales:

- Variables controlables. Son todas aquellas que pueden ser controladas o monitorizadas durante el proceso de experimentación. Por ejemplo, es una variable de este tipo, la potencia de transmisión de los nodos, ya que antes de la experimentación se puede fijar su valor, sabiendo con certeza que éste no variará durante el proceso experimental. Otros factores como éste son: la distancia de separación entre dispositivos, la localización de la red, la distribución del tráfico de paquetes, etc.
- Variables no controlables o aleatorias. Son esas que no pueden ser controladas o monitorizadas, y que por tanto adoptan un carácter aleatorio durante el proceso de experimentación. Fallos en el hardware de los dispositivos al tiempo que se llevan a cabo los tests o las condiciones climáticas, son algunos de esos fenómenos que están fuera del alcance de manejo por parte del usuario.

La excesiva cantidad de variables, tanto controlables como aleatorias, que influyen sobre los elementos de una red de sensores, nos obliga a estudiar en nuestra metodología sólo un reducido número de ellas. Éstas son: la distancia de separación entre dispositivos, la potencia de transmisión de los nodos, la localización de la WSN, la distribución del tráfico de paquetes, el protocolo de comunicación de la capa de transporte y el número de nodos sensores empleados en los experimentos. Son definidas a continuación, y a la vez se indicará que valor o rango de valores poseen para los diferentes experimentos de nuestra metodología.

### 2.2.1 Distancia

La distancia de separación entre nodos es uno de los parámetros más importantes que deben ser estudiados antes de desplegar una Red de Sensores Inalámbrica. Se debe experimentar con los futuros motes que constituirán la red para determinar, normalmente, cuál es el alcance máximo de la comunicación, es decir, cuál es la máxima distancia que podemos separar dos elementos de la red para que

se transmita y reciba información con el objetivo de cumplir los requerimientos funcionales y energéticos, y criterios de calidad mínimos del sistema.

Así pues, la distancia entre elementos es una de las variables de experimentación de la presente metodología, la cual es modificada en sucesivas pruebas con el fin de conocer como varían parámetros como la tasa de recepción de mensajes o la fuerza de la señal recibida. En principio, a mayor distancia de separación, peor deben ser las respuestas de parámetros como los dos citados anteriormente. De este modo, existirá una distancia a partir de la cual estos parámetros presenten unos valores que no cumplen los requerimientos y criterios de calidad mínimos de la red. El diseñador de la WSN establecerá entonces cualquier comunicación entre dispositivos de la red siendo ésa la distancia máxima de separación entre nodos.

También, en principio, cuanto mayor sea el alcance de comunicación, menor número de dispositivos deberán ser desplegados para cubrir toda el área que desea ser monitorizada. Una ventaja de esta medida sería la reducción de gastos en la adquisición de material de captación de fenómenos (especialmente importante en proyectos de despliegue con bajo presupuesto), aunque bien es cierto que motes con buenas características respecto al alcance puede conllevar a un aumento en su precio y por tanto desde el punto de vista económico mayor alcance en la comunicación no significaría un mayor ahorro. Otra ventaja clara de poseer nodos con un alto alcance sería la reducción del tráfico en la red, lo que significa menor número de paquetes en circulación y por tanto menor número de pérdidas o de errores en la comunicación.

Sin embargo, como desventaja, si disponemos de un alto máximo alcance, y por consiguiente desplegamos pocos elementos en la red, el número de enlaces posibles donde circularán los paquetes será pequeño. Esto acarrearía mayores problemas a la hora de redireccionar el tráfico si algún enlace falla, existiendo una menor combinación de posibles rutas alternativas.

A modo de resumen: tras realizar la experimentación con los dispositivos y obtener los datos suficientes, la decisión del diseñador sobre la distancia de separación final entre elementos de la red dependerá de esa información recogida en los tests y de los requerimientos del sistema.

## 2.2.2 Potencia de transmisión

Ejecutar un proceso experimental teniendo en cuenta la potencia de transmisión de los sensores es si cabe aún más importante que realizarlo con la variable distancia. Cada nodo de una Red de Sensores Inalámbrica necesita ser configurado para transmitir los mensajes a una potencia determinada. La elección de dicha potencia también depende de los requerimientos funcionales y de calidad del sistema, pero principalmente la decisión se realizará atendiendo a las restricciones energéticas, que son las que tienen una mayor importancia, ya que marcan el tiempo de vida de la red.

Como suposición inicial se puede establecer que para una distancia fija, cuanto mayor es esta variable, mejores serán ciertos parámetros como por ejemplo la tasa de recepción de mensajes. De este modo a mayor potencia de transmisión de paquetes, en principio los requerimientos de la red se verán cumplidos para una mayor distancia de separación de los nodos.

Otra suposición inicial, la cual es crucial, es su relación con la batería de los sensores. Tal y como se tratará de demostrar en el Experimento C, cuando esta variable se incrementa, el gasto energético de ve afectado. Tratará de demostrarse en la ejecución del experimento sobre un juego de sensores comerciales. Como decimos esta relación potencia-batería es vital y por tanto debe ser estudiada minuciosamente por el diseñador de la WSN antes de que el sistema sea diseñado.

## 2.2.3 Localización

El lugar donde la red de sensores es situada es una variable que viene estrechamente ligada al desempeño funcional que el sistema va a realizar. En [1] y [2] se demuestra que según la localización de la red inalámbrica, existen diferencias significativas en el comportamiento de los elementos. De este modo, para cada área distinta donde quiera desplegarse una red de sensores, se necesitará llevar a cabo un estudio de evaluación del comportamiento de los dispositivos en esa localización concreta.

Esta variación en el comportamiento de los motes según el lugar donde se sitúen tiene su explicación en que para cada localización, los fenómenos y variables que afectan la

comunicación inalámbrica cambian. Así, por ejemplo, en aplicaciones para la monitorización de cultivos o comunidades de animales, las redes serán desplegadas normalmente en una localización de exterior, la cual se verá afectada intensamente por las condiciones climáticas, y en un terreno con posible presencia en abundancia de vegetación, tierra o rocas. Sin embargo, otro tipo de aplicaciones, como el control de incendios en edificios, requerirían un despliegue de la red en una localización de interior, donde el clima no sería un factor determinante y donde la superficie sería muy distinta. Estos factores pueden ayudar en mayor o menor medida por ejemplo a atenuar la señal o en cambio a facilitar la comunicación inalámbrica.

En nuestra metodología se definen a continuación tres entornos diferentes. En un principio el principal objetivo era ejecutar los experimentos en un entorno de exterior y otro de interior. Por ello, se seleccionó un terreno con gran cantidad de vegetación como lo es la parcela de césped escogida, y un pasillo de gran longitud en el interior de un edificio. Actualmente, ambas localizaciones soportan un gran número de ejemplos de aplicaciones de las WSNs. Finalmente, se seleccionó también una segunda área en el exterior (un parking de asfalto para automóviles) con el fin de establecer diferencias en los resultados de los experimentos.

#### ■ **Pasillo.**

Actualmente una gran cantidad de aplicaciones sobre WSNs se ejecutan en localizaciones de interior, tal como en edificios de viviendas, oficinas, almacenes, etc. Algunos ejemplos serían: un sistema de localización de ancianos en residencias o la monitorización de los productos existentes en un almacén. Por ello es necesario que en nuestra metodología de experimentación se especifiquen los pasos a seguir para testear los elementos sensores en este tipo de superficie.

Los experimentos, cuyos resultados se muestran en el Capítulo 3, se han ejecutado en un pasillo de un edificio de oficinas con una longitud de 45 metros y una anchura de 2 metros (ver la Figura 2.1). Debido a su gran estrechez, será importante evaluar los posibles efectos de los muros, ya que el fenómeno multi-camino podría ser de gran influencia para la mejora de las prestaciones en la comunicación inalámbrica.



**Figura 2.1** Pasillo donde se han ejecutado los experimentos

■ **Terreno de césped.**

Las aplicaciones llevadas a cabo en superficies con vegetación son muy numerosas. Una característica muy importante de este tipo de localizaciones es el hecho de que al ser normalmente en el exterior, los elementos que forman la red se encuentran expuestos a las diferentes condiciones climáticas, las cuales pueden modificar su rendimiento, añadiendo en ocasiones un carácter aleatorio a los resultados obtenidos en la experimentación. En concreto, se ha completado la experimentación sobre una parcela de césped rigurosamente rectangular con dimensiones de 50 x 50 metros (ver la Figura 2.2).



**Figura 2.2** Parcela de césped donde se han ejecutado los experimentos

■ **Parking de asfalto.**

Una tercera ejecución de los experimentos es llevada a cabo en otra área de exterior sobre un parking para automóviles cuyas dimensiones son 50 metros de longitud y 10 metros de anchura (ver la Figura 2.3). Aplicaciones como un sistema para el control de semáforos y barreras, puede ser un ejemplo de uso de Redes de Sensores Inalámbricas sobre este tipo de superficie.



**Figura 2.3** Parking de asfalto donde se han ejecutado los experimentos

## 2.2.4 Distribución del tráfico de paquetes

El flujo de datos que un nodo de una red transmite y recibe debe ser estudiado y caracterizado. Tal y como se indicó en el capítulo de introducción a las WSNs, existen tres principales modos de uso de este tipo de redes: monitorización sincronizada, monitorización en base a eventos y monitorización para seguimientos. En estos dos últimos usos, los nodos de la red envían información en un tiempo aleatorio atendiendo al momento en que se lleven a cabo los eventos o acorde a los cambios en los elementos que están siendo rastreados. Sin embargo, en las aplicaciones que funcionan con monitorización sincronizada, los nodos de la red, tras realizar la captación de los fenómenos, envían los datos en un tiempo definido y de forma periódica. Estos datos son transmitidos a otros nodos o a la pasarela, los cuales se encargan de reenviarlos a la estación base, donde es analizada la información. Por tanto, esos datos enviados en un tiempo específico y conocido (ventana temporal) a

través de la red, se corresponden con un número determinado de paquetes fijo. Así pues, de antemano se conoce la cantidad de información que va a viajar a través del sistema. Se dice que el tráfico de paquetes sigue una distribución *uniforme*. Los nodos con los que se experimenta en el presente trabajo están configurados para enviar paquetes siguiendo este tipo de distribución.

## 2.2.5 Protocolo de la capa de transporte

Las Redes de Sensores Inalámbricas están conectadas normalmente, gracias a las pasarelas, a una red de mayor tamaño como Internet. De aquí radica la necesidad de la existencia de una capa de transporte en las WSNs.

Los dos protocolos de comunicación de la capa transporte más populares son TCP [19] y UDP [20]. En principio, las WSNs podrían trabajar con ellos, pero en realidad estos protocolos presentan numerosos problemas al trabajar con dispositivos de reducidos recursos como lo son los nodos sensores. C. Wang y K. Sohraby [21] desvelan las diversas limitaciones de trabajar con TCP y UDP en WSNs y presentan el diseño de diversos protocolos de transporte alternativos. Además, P. Pereira y A. Grilo, entre otros autores, también afirman en [22] que TCP y UDP no son adecuados para trabajar en redes de sensores, presentando a la vez las pautas para el diseño de protocolos que si operarían de forma correcta en este tipo de sistemas.

En algunos otros trabajos se puede encontrar que es factible trabajar sobre WSNs con un TCP modificado. Sin embargo para redes de pocos dispositivos se aconseja trabajar con UDP mejor que con TCP. Existen tres motivos principalmente:

1. TCP es un protocolo orientado a conexión y de gran fiabilidad que obliga a los dispositivos a llevar un exhaustivo control en la comunicación con el almacenamiento de información de estado. Ésta no es una buena solución para nodos que tienen una importante carencia de recursos de memoria.
2. Al contrario que UDP, TCP realiza control de congestión. Sin embargo, en sistemas con un reducido número de dispositivos esta técnica no es de gran utilidad. Además, en los sistemas de monitorización sincronizada donde los datos son enviados de forma periódica, el control de congestión es aún más inservible, ya que el administrador de la red puede planificar los tiempos de adquisición de



datos en los nodos con el fin de no causar cuellos de botella en las pasarelas.

3. TCP es un protocolo fiable que contiene sistema de reconocimiento de llegada de los mensajes al destino. Sin embargo, este mecanismo generalmente no es adecuado para las Redes de Sensores Inalámbricas, ya que cuando los paquetes no llegan al destino, estos tienen que ser retransmitidos, originando un aumento del tráfico en la red y por tanto mayor consumo energético en los nodos. En cambio, UDP no implementa esta técnica.

Estas tres razones y el hecho de que en la experimentación de este proyecto se utiliza un número de nodos sensores reducido, convierten al protocolo UDP, en comparación con TCP, en el mecanismo ideal para llevar a cabo la comunicación entre los nodos en el nivel de transporte, aún sabiendo que actualmente existen técnicas en este nivel mejores para trabajar en WSNs.

## 2.2.6 Número de nodos

Cuando una red de sensores va a ser diseñada, antes de adquirir todo el conjunto de dispositivos que la constituirán, es imprescindible realizar el proceso de caracterización de los nodos con el fin de comprobar si dichos sensores cumplen los requerimientos, características y criterios de calidad del sistema. Por consiguiente, al principio, sólo un reducido número de elementos del tipo seleccionado, debería ser adquirido, evitando de este modo, un fuerte sacrificio económico en el caso de que dichos dispositivos no cumplan los requisitos que la red requiere. En ese caso, se debería obtener un nuevo juego de dispositivos para realizar el mismo proceso experimental.

Por tanto, los experimentos de esta metodología han sido diseñados para ser ejecutados únicamente con 2 nodos sensores (1 emisor y 1 receptor), a excepción del Experimento B en la parcela de césped, donde se utilizan 5 dispositivos (1 emisor y 4 receptores). Esta decisión reduce el rigor del conjunto de experimentos desarrollados en esta metodología, ya que normalmente las WSNs contienen muchos más elementos que transmiten y reciben mensajes a y hacia otros nodos. Sin embargo, finalmente se ha utilizado tal reducida cantidad de sensores debido a que se disponía de un juego de nodos escaso, y también a que el estudio de las colisiones añade una gran dificultad al realizar el análisis de los resultados.

La metodología para la caracterización de sensores inalámbricos ha sido dividida en tres experimentos atendiendo a las diferencias en la configuración del escenario, la disposición de los elementos y los parámetros de los sensores testeados. Así, en el Experimento A se estudia la tasa de recepción de paquetes, la fuerza de la señal recibida y la variación del indicador de calidad del enlace. Por otro lado en el Experimento B se analiza el comportamiento de los nodos según su ángulo de orientación, mientras que en el C se estudia la respuesta energética de los dispositivos, es decir, la vida de sus baterías.

Debido a que en cada experimento se testean ciertas características o parámetros diferentes, en cada uno de ellos se debe emplear también una configuración distinta, y por tanto algunas de las variables experimentales tienen un valor diferente para cada uno de ellos. Los valores de los factores para cada experimento son resumidos en la Tabla 2.1.

Experimento	Distancia	Potencia Tx	Localización	Distribución	Protocolo	Nº Nodos
<b>A</b>	Variable	Variable	Todas	Uniforme	UDP	2
<b>B</b>	Variable	Fija	Todas	Uniforme	UDP	2 / 5
<b>C</b>	Fija	Variable	Sólo Pasillo	Uniforme	UDP	2

**Tabla 2.1** Valores de los factores para cada experimento

## 2.3 Experimento A

Mediante la ejecución de este experimento se recogen los datos suficientes para analizar la tasa de recepción de paquetes del dispositivo, cómo varía la fuerza de la señal recibida por un nodo receptor o cómo lo hace el indicador de calidad del enlace. Todos ellos son estudiados mediante la ejecución del mismo experimento, ya que para cada uno se emplea idéntica configuración del escenario y los elementos (mismo software). Además, los tres factores se estudian mientras la distancia de separación y la potencia de transmisión son variadas en cada una de las localizaciones descritas anteriormente. Sin embargo, la distribución del tráfico de paquetes, el protocolo de la capa de transporte y el número de nodos empleados son fijos.

En las siguientes subsecciones se desarrolla un análisis teórico de los tres factores testeados en este experimento y finalmente se explica paso por paso detalladamente el proceso para llevar a cabo la experimentación de esas características de los nodos y la comunicación.

### 2.3.1 Tasa de recepción de paquetes

Uno de los parámetros más importantes en toda comunicación inalámbrica es la tasa de recepción de paquetes de los elementos de la red. Cualquier sistema de telecomunicación esta constituido por componentes que generan cierto tráfico de datos, el cual puede ser exitosamente entregado al destino o por el contrario puede perderse o descartarse. De este modo, la *tasa de recepción de paquetes* es definida como la fracción de paquetes enviados y recibidos correctamente por el destino en una ventana temporal concreta. Además, en relación a ese parámetro se puede definir la *tasa de pérdida de paquetes* como la fracción del tráfico transmitido el cual no es entregado correctamente en cierto intervalo de tiempo. La suma de ambas variables determinaría el tráfico total circulante por la red.

En el caso de que se utilicen protocolos que no contienen servicio de retransmisiones o que simplemente no aseguran de un modo fiable la entrega de la información, tal como UDP, el tráfico no recibido puede representar una cantidad porcentual importante respecto al total de tráfico que circula por la red. Esto supondría un problema de gran magnitud para aplicaciones donde la recolección de la información es vital, como por ejemplo cualquier sistema de seguridad antirrobo o prevención de incendios. Para dichas WSNs la tasa de pérdida de paquetes debe tender a 0.

En cambio, existen otras muchas aplicaciones donde la entrega de la información adquirida por los sensores no es tan crucial. Por ejemplo: en la monitorización de un campo de cultivo, la pérdida de algún paquete el cual informa de la humedad, no representaría un gran problema para el desempeño funcional final del sistema. Otro ejemplo sería el de un sistema que trabaje con voz sobre IP. La pérdida de algún paquete significaría una pequeña distorsión vocal, que no entraña para la mayoría de estas aplicaciones ningún problema crítico. En sistemas de este estilo se suele establecer un umbral para parámetros como este. Así, en los requerimientos de diseño de la red de sensores se podría estipular a cumplir una tasa de recepción de paquetes mínima del 95%. Por tanto, el diseñador debería realizar la caracterización de los

nodos adquiridos en el escenario real donde trabajará la red para asegurar que dicho requisito se cumple. Para ello, se podría completar una metodología como la definida en este proyecto.

Se trata pues de un parámetro, cuyo estudio y conocimiento es de gran importancia para cualquier Red de Sensores Inalámbrica. Por este motivo ha sido incluido en nuestra metodología.

Mediante el Experimento A se va a testear la tasa de recepción de paquetes de un nodo sensor mientras se varía la potencia de transmisión de un emisor y la distancia de separación entre transmisor y receptor. Existen autores que han realizado trabajos con el fin de estudiar este parámetro. Algunos ejemplos son:

- J. Zhao y R. Govidan presentan en [2] una metodología para estudiar la tasa de recepción de los sensores en una WSN. De los resultados obtenidos en el estudio, se afirma que dicha tasa aumenta conforme lo hace la potencia de transmisión y disminuye a la vez que se incrementa la distancia de separación entre dispositivos. Además, se demuestra que el comportamiento de los nodos es bastante distinto para cada uno de los entornos donde se ejecuta la metodología. Por último, también se comenta la existencia de “áreas grises”, donde se divisan irregularidades en la comunicación.
- A. Cerpa y J.L. Wong, entre otros autores, desarrollaron un estudio [23] donde para nodos de WSNs se analizaba, entre otros parámetros, la relación de la tasa de recepción con la distancia. Así, se afirmaba que generalmente esta tasa disminuía al aumentarse la separación entre dispositivos. Pero su hallazgo más interesante fue la demostración de la existencia de irregularidades en la comunicación, ya que para determinados alcances fijos, se obtenían probabilidades similares de obtener una muy alta o una muy baja tasa de recepción. Además, demostraron que incluso para distancias de separación muy cortas, la probabilidad de obtener una muy baja tasa de recepción no es cero.

Acorde a estos trabajos relacionados con el estudio de la tasa de recepción en nodos de redes de sensores y también según algunas suposiciones teóricas establecidas previamente, después de ejecutar la experimentación y analizar los resultados obtenidos, se espera obtener las siguientes afirmaciones:

- La tasa de recepción en un nodo crece si la potencia de transmisión es incrementada, para una distancia de separación fija.
- Estableciendo una potencia de transmisión fija, al aumentar la distancia entre dispositivos, la tasa de paquetes recibidos por el nodo disminuye.
- El comportamiento de los nodos sensores respecto a este parámetro es diferente para cada una de las localizaciones donde se ejecuta la experimentación.
- Existencia de grandes irregularidades en la comunicación inalámbrica entre los elementos.

Tras la ejecución de la metodología sobre un juego de sensores comercial, en el Capítulo 3 se comprobará el cumplimiento de estas afirmaciones tras analizar los resultados obtenidos. Además se indicarán las diferencias para este parámetro en cada uno de los entornos y se comprobará si existe una relación de proporcionalidad entre el nivel de potencia transmitida y el alcance máximo de la comunicación.

### 2.3.2 Fuerza de la Señal Recibida

La *Fuerza de la Señal Recibida* (Received Signal Strength, RSS) es una medida de la potencia de la señal capturada por un nodo receptor. Se corresponde con un parámetro de gran utilidad cuando se desea estudiar las prestaciones de un sistema de comunicación.

Una vez se conoce su valor y el de la potencia de transmisión, el RSS puede ser usado para averiguar la atenuación que existe en el canal de comunicación. También se podría utilizar para averiguar la sensibilidad del nodo receptor para ser capaz de captar información, es decir, que potencia de señal mínima debe llegar a un determinado nodo para que éste reciba el paquete. En nuestra metodología se emplea el dBm como unidad de medida del parámetro RSS.

En trabajos como los llevados a cabo por M. Mallinson, P. Drane y S. Hussain en [24] o S. Lin y J. Zhang en [1], se estudia la respuesta del parámetro RSS en función de la

distancia y la potencia de transmisión. Las medidas recogidas muestran resultados en principio triviales, tales como la reducción de la fuerza de la señal con el incremento de la distancia o su aumento cuando la potencia de transmisión crece. Sin embargo, en las gráficas de resultados se aprecia que la dependencia de este parámetro con la distancia de separación no es lineal, decayendo la fuerza de la señal en mayor medida en los primeros metros de la escala de experimentación.

En el capítulo de resultados se comprueba si todas estas relaciones se verifican, y además se comentan otros aspectos apreciados sobre los datos recogidos. La verificación de estas relaciones puede ser un primer paso de demostración de que la presente metodología de caracterización funciona correctamente y por tanto es del todo válida para su uso.

### 2.3.3 Indicador de Calidad del Enlace

El *Indicador de Calidad del Enlace* (Link Quality Indicator, LQI) es una métrica de la calidad de la señal de entrada. Cuando un nodo lo implementa, por cada mensaje recibido se mide la tasa de error de los ocho símbolos primeros del paquete. Este parámetro es usado como una medida relativa de la calidad del enlace (un valor alto indica mejor enlace que uno bajo), ya que el valor es dependiente del formato de modulación.

Existen 5 “casos extremos” que pueden ayudar a ilustrar el significado de los parámetros RSS y LQI:

1. Una señal débil que se transmite por un canal con presencia de ruido puede provocar que se mida en recepción un valor bajo tanto para el RSS como para el LQI.
2. Una señal débil que se envía por un enlace donde no existe absolutamente nada de ruido, proporciona un valor bajo de RSS y uno alto para el LQI.
3. Un fuerte ruido (normalmente procedente de un sistema que interfiere) puede dar un valor alto para el RSS y bajo para el LQI.
4. Una señal fuerte que se transmite sin la presencia de mucho ruido puede

originar la medición de un alto RSS y un alto LQI.

5. Una señal muy fuerte que causa la saturación del receptor puede propiciar un alto valor para el RSS y bajo para el LQI.

Al igual que con el parámetro de la Fuerza de la Señal Recibida, los autores de los trabajos [1] y [24], estudiaron y analizaron los resultados de las mediciones del Indicador de Calidad del Enlace. Acorde a esas investigaciones, este parámetro decrece cuando se aumenta la distancia de separación entre dispositivos y también lo hace cuando se disminuye la potencia de transmisión. Además, para algunos de los entornos se cumple que en cortas distancias el valor del LQI no cambia aunque lo haga la potencia de transmisión.

De la misma forma que para los otros dos parámetros del Experimento A, los anteriores hallazgos de otros autores trataran de ser confirmados en el Capítulo 3 de resultados, tras la ejecución de la metodología.

### 2.3.4 Procedimiento

El siguiente proceso debe ser estrictamente completado para alcanzar el objetivo de conseguir una correcta ejecución del Experimento A, y de este modo conocer el comportamiento de los sensores testeados para las tres características explicadas anteriormente. El procedimiento es descrito paso por paso.

1. Cargar las distintas aplicaciones software en los dispositivos transmisor y receptor. Por un lado, el programa del emisor debe crear 20 paquetes UDP y enviarlos con una tasa de 1 paquete por segundo (distribución de tráfico uniforme). Este nodo sensor debe ser configurado para que dichos paquetes puedan ser enviados por el usuario un número de veces infinitas. Por ejemplo, esto se podría realizar con la configuración de un interruptor que activara el envío. Además, el software debería ser desarrollado de modo que se le permita al usuario controlar el nivel de la potencia de transmisión.

Por otro lado, la aplicación en el receptor debe ser programada para capturar y contar los paquetes que vienen de la fuente. Además, se debe almacenar los valores de los parámetros RSS y LQI para cada paquete que se recibe (Por

ejemplo en estructuras de datos como los vectores o arreglos). Por último, se aconseja configurar ambos dispositivos para que diferentes LEDs respondan a diferentes eventos, como la recepción de un paquete, con el fin de servir de ayuda en el proceso de experimentación.

2. Fijar con cinta adhesiva ambos sensores a diferentes cajas de cartón con el fin de evitar movimientos que tergiversen el experimento. Las cajas deben ser lo suficientemente anchas y largas para que los sensores puedan estar situadas sobre ellas de forma estable. Además, serán lo suficientemente altas como para que ningún objeto (por ejemplo el césped) sea un obstáculo en el enlace directo entre el emisor y el receptor. Las cajas seleccionadas para nuestro experimento tienen un tamaño de 330 mm de largo, 200 mm de ancho y 75 mm de alto.
3. Las medidas son recogidas en los tres entornos: pasillo, parking y terreno de césped; utilizándose distintas herramientas para crear la escala métrica en cada localización:

Pasillo: se hacen pequeñas marcas cada 2 metros sobre el suelo o la pared (con cinta adhesiva por ejemplo) a lo largo de todo el pasillo. Estas indican donde se sitúa el nodo receptor en cada iteración (ver la Figura 2.4-a).

Parking: se realizan las marcas sobre el asfalto por cada metro. Para ello se utilizará por ejemplo tiza o pintura blanca. Debido a que la comunicación inalámbrica puede presentar en principio en las localizaciones exteriores peor prestaciones que en el pasillo, se debe pintar marcas para las distancias de 0.5 y 1.5 metros (también marcarlas para la escala del terreno de césped), con el fin de obtener mayor precisión en nuestras medidas. Este es también el motivo por el que la escala en los dos entornos exteriores es cada metro en lugar de cada dos (ver la Figura 2.4-b).

Césped: se utiliza una cuerda de un grosor aproximado de 2 milímetros y tan larga como se desee la escala (por ejemplo 25 metros). Los extremos se deben atar cada uno a la cabeza de un clavo, los cuales son clavados en el terreno estando la cuerda totalmente tensa. Cada metro de la cuerda se fija cinta adhesiva para crear la escala. No utilizar nunca otros clavos para esta tarea, ya que si son metálicos pueden influir de manera notoria en el



experimento. Así, la escala numérica será similar a la creada en el parking de asfalto (ver la Figura 2.4-c).

4. La ejecución del experimento en si comienza en esta fase, la cual es común para cada entorno. El nodo emisor debe estar situado sobre su caja al comienzo de la escala métrica, estando posicionado el receptor en la primera marca de dicha escala (2 metros en el pasillo y 0.5 metros en el parking y el césped), ambos sobre el mismo plano y con la misma orientación. Entonces, activar la transmisión de la información y esperar hasta que los 20 paquetes son enviados. A continuación, mover el nodo receptor a la siguiente distancia de la escala y activar el envío de nuevo. Este proceso se debe ejecutar hasta que se complete la escala métrica.
5. Extraer los datos almacenados en el sensor receptor y guardarlos.
6. Ejecutar las fases 4 y 5 dos veces más para así poder obtener una mayor cantidad de medidas y por tanto nuestros posibles futuros hallazgos sean más rigurosos y fiables.
7. Variar el valor de la potencia de transmisión en el emisor y completar las fases 4, 5 y 6.

Finalmente, por cada potencia de transmisión se obtienen 3 ejecuciones o ciclos completos de la escala métrica. Seguidamente la información recogida sobre los tres parámetros testeados debe ser estudiada con el fin de poder obtener conclusiones sobre el comportamiento de los nodos sensores estudiados.



a) Pasillo

b) Parking de automóviles

c) Césped

**Figura 2.4** Disposición de los elementos en las distintas localizaciones.

## 2.4 Experimento B

El objetivo de este experimento es obtener una representación de la respuesta en la comunicación de los nodos sensores cuando se modifica su orientación. En concreto, se analiza su comportamiento en base al ángulo de orientación mediante el testeo de la tasa de recepción de paquetes, de la Fuerza de la Señal Recibida y del Indicador de Calidad del Enlace. El experimento es ejecutado en los tres entornos para una potencia de transmisión fija, mientras que la distancia de separación entre los elementos es variable.

### 2.4.1 Orientación de los nodos

El alcance máximo es un parámetro que permite al diseñador de la red de sensores obtener una idea de la cantidad de dispositivos que debe utilizar y donde ellos deben situarse a lo largo del área por la que se desplegará la WSN. Sin embargo, se debe determinar también la posición y orientación exacta de cada uno de los elementos de la red con el fin de mejorar en la mayor medida posible, la comunicación entre dispositivos, para así optimizar el desempeño funcional del sistema. Así, el *ángulo de orientación* que deben tener los nodos de una red debe ser minuciosamente estudiado con el objetivo de obtener las mayores prestaciones posibles para la WSN.

En la mayoría de los casos existen ciertas direcciones mediante las cuales un nodo sensor transmite con mayor potencia que otras. Este hecho no sólo depende de las características teóricas de la antena incorporada en el mote, sino también de otros muchos factores que convierten en una ardua tarea el llevar a cabo la definición detallada del diagrama de radiación real que la antena presenta cuando el dispositivo es situado en un entorno específico. Por este motivo, aunque sea conocido el comportamiento teórico de la antena de los nodos, se debe experimentar con éstos sobre el escenario real donde la red inalámbrica será desplegada.

Según los requerimientos exigidos por la red, se deben analizar distintos parámetros característicos de los sensores según sus ángulos de posicionamiento. Así por ejemplo, se puede preferir llevar a cabo la orientación de los nodos con el objetivo de minimizar los paquetes perdidos en la red. En este caso se debe estudiar la tasa de recepción para cada posición entre el emisor y el receptor. En cambio, si interesa minimizar el error en la información recibida, se tiene que realizar un test sobre el

Indicador de Calidad del Enlace para cada una de las orientaciones posibles de los nodos.

En el Experimento B se ejecuta un test para evaluar las características de las distintas posiciones entre un nodo emisor y un nodo receptor. Para ello se recogen medidas de la tasa de recepción de paquetes, la Fuerza de la Señal Recibida y del Indicador de Calidad del Enlace. Sería aconsejable completar este experimento antes del Experimento A, para que de esta forma se puedan recolectar las medidas una vez conocidas las orientaciones óptimas de los sensores y de este modo se pueda obtener, por ejemplo, el verdadero alcance máximo de la comunicación.

## 2.4.2 Procedimiento

El procedimiento llevado a cabo para la ejecución del Experimento B es muy similar al conjunto de pasos descritos para el Experimento A. Las diferencias más notables son la introducción de tres receptores más en el experimento en el terreno de césped y el valor fijo de la potencia de transmisión.

1. Se pueden cargar en los sensores los mismos programas utilizados en el Experimento A. La potencia de transmisión puede tener cualquier valor, aunque éste debe ser el mismo para todas las posiciones de los sensores testeadas. Sin embargo, el nivel de potencia seleccionado no debe ser demasiado bajo, con el objetivo de recolectar una significativa suma de medidas para cada orientación, ya que si la potencia es débil, el alcance de la comunicación podría ser bastante corto (principalmente en los entornos de exterior) y por tanto se recogerían menos datos.
2. Los sensores deben ser fijados a sus respectivas cajas, las cuales tienen las mismas dimensiones que las empleadas en el experimento previo. En el caso del terreno de césped, se emplean 3 cajas más para tantos nuevos receptores.
3. Las escalas métricas también son creadas como en el Experimento A, excepto en el terreno de césped, donde dos cuerdas de aproximadamente 50 metros cada una tienen que ser utilizadas. Atar los extremos de las cuerdas a cuatro clavos y clavar éstos en el terreno dejando las cuerdas muy tensas. Ello debe hacerse de modo que las cuerdas formen una cruz de 4 ángulos rectos con sus

centros unidos y se obtenga por tanto cuatro escalas diferentes de 25 metros cada una. Utilizar cinta adhesiva para realizar las marcas cada metro en cada una de las escalas.

4. Existen sustanciales diferencias en la ejecución del experimento entre el caso del terreno de césped y el resto de localizaciones:

Pasillo/Parking: el transmisor y el receptor deben ser situados al comienzo de la escala métrica y sobre la primera marca de testeo respectivamente. Ambos se tienen que colocar sobre el mismo plano y con el mismo ángulo de orientación, definido como  $0^\circ$ .

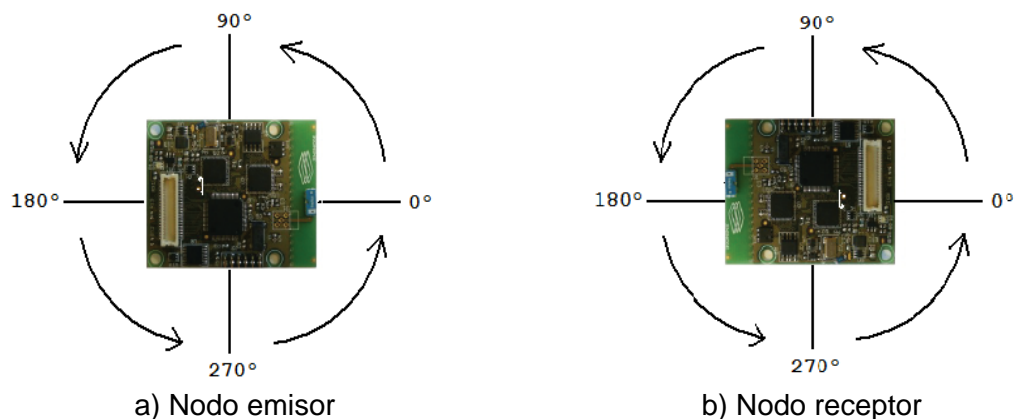
Césped: el emisor es colocado en la unión de los centros de las cuerdas, mientras que los receptores se sitúan sobre la primera marca de las 4 escalas. Todos los sensores deben estar sobre el mismo plano y con el mismo ángulo de orientación de  $0^\circ$ .

A continuación, se activa la transmisión de los 20 paquetes UDP y cuando se hayan enviado, los receptores deben ser movidos hacia la siguiente distancia de testeo. Repetirlo hasta que finalicen las escalas métricas.

5. Extraer la información almacenada en la memoria de los nodos sensores.
6. Las fases 4 y 5 deben ser ejecutadas 2 veces más.
7. En esta fase se realiza el cambio de orientación de los sensores y a continuación se ejecuta el experimento de nuevo, es decir, se completan otra vez las fases 4, 5 y 6. El proceso también es distinto según la localización:

Pasillo/Parking: siempre sobre el mismo plano, rotar  $90^\circ$  el sensor transmisor fijado a su caja y realizar las fases 4, 5 y 6. Entonces, rotarlo  $90^\circ$  otra vez y repetir de nuevo esas 3 fases. Llevar a cabo este proceso hasta que el emisor vuelva a su posición inicial de  $0^\circ$ , es decir, hasta que realice una vuelta completa. A continuación, rotar el receptor  $90^\circ$  y repetir el experimento rotando el transmisor  $90^\circ$  en cada iteración (4 posiciones:  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$  y  $360^\circ$ ). Finalmente, las fases 4, 5 y 6 son completadas 16 veces, siendo la orientación de los dispositivos diferente para cada una de ellas. La

Figura 2.5 muestra un ejemplo de comunicación entre ambos nodos, con el emisor en la posición de  $0^\circ$  y el receptor en  $180^\circ$ . Como se aprecia, este último está rotado  $180^\circ$  sobre el mismo plano.



**Figura 2.5** Ejemplo de colocación de los nodos sensores para el Experimento B

Césped: en esta área el test sobre la orientación de los sensores es más sencillo, ya que al emplearse 3 receptores más, se debe repetir un menor número de veces el experimento. Al contrario que en las otras localizaciones, los receptores no son rotados en ningún momento. Solamente el emisor se rota  $90^\circ$  cada vez que las fases 4, 5 y 6 son completadas, y por tanto finalmente se obtienen 4 conjuntos de medidas (uno para cada posición del transmisor), aunque son 16 las diferentes orientaciones testeadas.

Por tanto, solamente se emplea un único receptor en el pasillo y en el parking de asfalto debido a que sus dimensiones no permiten ejecutar un experimento con las escalas en forma de cruz. Por otro lado, finalmente para cada una de las localizaciones se obtienen 16 diferentes posiciones entre los nodos testeadas, lo que permite obtener una amplia definición del comportamiento de los dispositivos sensores respecto a sus ángulos de orientación en el área.

## 2.5 Experimento C

La ejecución del Experimento C y el posterior análisis de los datos recogidos permiten averiguar cual es el comportamiento de los nodos sensores testeados respecto a al gasto energético. En este experimento se testea la vida de la batería de

los dispositivos en función de tres parámetros: los paquetes enviados, los paquetes recibidos y la potencia de transmisión. En el siguiente subapartado se explica en profundidad la importancia de la energía en las redes de sensores, y posteriormente se describe detalladamente el procedimiento para ejecutar el Experimento C.

### 2.5.1 Vida de la batería

La característica inalámbrica de los sensores que forman una red y su necesario pequeño tamaño, el cual les limita la disposición de más recursos, les obligan a tener serías restricciones respecto al consumo de energía. Así pues, debido a que no pueden almacenar una gran cantidad de ella, la vida del dispositivo y por tanto la vida de la WSN dependen completamente de como la batería es descargada.

Prolongar la vida de la batería de un mote es el principal objetivo en el diseño de Redes de Sensores Inalámbricas debido a la dificultad y al alto coste asociado al reemplazo o a la recarga de las baterías en una red ya desplegada. Por este motivo, actualmente existe una gran suma de investigaciones que tienen como objetivo el desarrollo de técnicas y tecnologías que minimicen el consumo de energía en las WSNs.

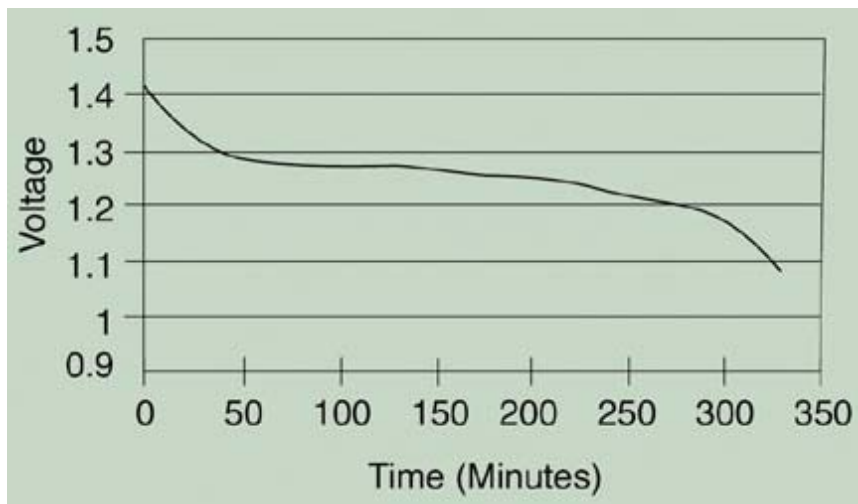
Por otro lado, el hardware del nodo sensor debe ser diseñado siempre desde el punto de vista del mínimo consumo. Además, el microcontrolador, el sistema operativo y el software cargado en el dispositivo deben optimizar la conservación de energía, inclusive llegando al extremo de escoger las instrucciones y su codificación, para que las secciones de código más utilizadas empleen la menor potencia posible.

Por consiguiente, probablemente *la vida de la batería* es el parámetro más crítico de todos aquellos que determinan el comportamiento de una WSN y de ahí la necesidad de realizar un minucioso estudio experimental sobre el consumo energético de un determinado juego de nodos sensores, lo cual es llevado a cabo en la presente metodología.

Como se ha indicado, dada la importancia de este parámetro, existen numerosos estudios donde se testea el consumo de la batería de los nodos en función de diversos factores, tales como el intervalo de muestreo (tiempo entre dos transmisiones consecutivas), la temperatura ambiente, el tiempo de transmisión (tiempo total que el

dispositivo está transmitiendo por intervalo de muestreo) o la potencia de transmisión. Este último factor es testado en trabajos como el llevado a cabo por C. Park, K. Lahiri y A. Raghunathan en 2005 [25], donde demuestran que la batería se descarga de forma más rápida si la potencia de transmisión es incrementada. Además, otros autores como S. Blom y C. Bellettini [4], determinan en su estudio que no existe una relación lineal en ese decrecimiento de la energía almacenada con respecto a la potencia que se emite.

Por otro lado, nuestra metodología es ejecutada sobre un juego de nodos comercial que utilizan baterías NiMH. De forma general, la relación entre el tiempo y el voltaje (energía) para este tipo de baterías tiene el aspecto mostrado en la Figura 2.6. Se comprobará tras la ejecución del Experimento C, si los datos recogidos reflejan la misma característica que la observada en esta figura.



**Figura 2.6** Relación entre la vida de la batería NiMH y su voltaje

El modo en que afecta a la vida de la batería la variación de la potencia de transmisión es una de las relaciones que estudia nuestra metodología. De modo que se comprobarán, en el capítulo de resultados, las afirmaciones anteriores, extraídas de diversos trabajos de investigación. Además el presente proyecto estudia también la relación de la energía almacenada en los nodos con respecto al número de paquetes enviados y al número de paquetes recibidos.

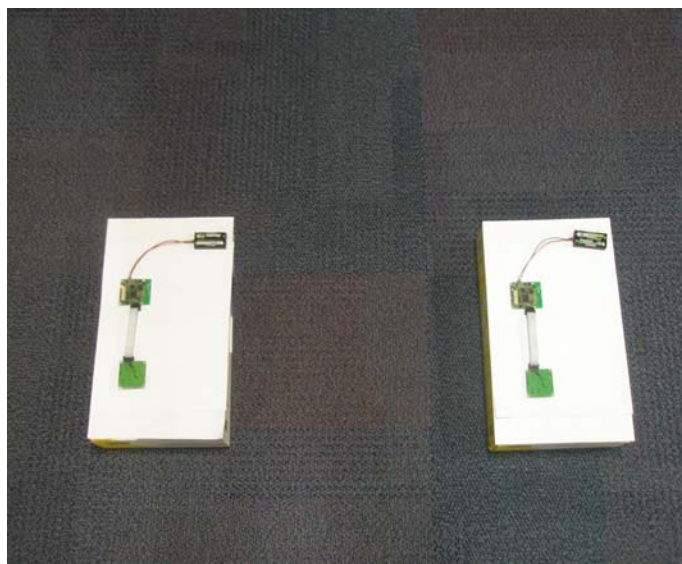
## 2.5.2 Procedimiento

Antes de testear la vida de la batería de los nodos según la potencia de transmisión, los paquetes enviados y los paquetes recibidos, se debe ejecutar un test inicial donde no intervengan esos tres factores. En este test sólo utilizaremos un sensor, cuya aplicación software cargada en él debe ejecutar el mismo código que los programas que se cargarán en el emisor y receptor en el experimento principal, pero excluyendo la tareas de envío y recepción de paquetes.

En este experimento inicial, se deben tomar medidas de la batería con un voltímetro cada 15 minutos durante las dos primeras horas, y cada 30 minutos durante el resto del experimento, ya que el nivel de la batería decrece más rápido al comienzo, tal y como se puede ver en la Figura 2.6. De este modo, al ser el intervalo más pequeño para las primeras muestras, las medidas tomadas son más precisas y definen mejor la función del voltaje respecto al tiempo. Después de este test inicial, se debe completar el siguiente proceso:

1. Cargar las aplicaciones software en el transmisor y en el receptor. Estas deben ser programadas para que se envíen 10 paquetes por segundo de un nodo a otro. Además, en el emisor una variable debe contar los paquetes enviados, al igual que en el receptor otra tiene que hacer lo propio con los mensajes capturados. También, el transmisor debe enviar los 10 pps durante un tiempo indefinido y se debe programar un mecanismo para cambiar el nivel de la potencia de transmisión.
2. Poner los dos sensores inalámbricos sobre sus cajas (con las mismas dimensiones que en los experimentos anteriores) y situarlos sobre el suelo en el pasillo, sólo a 0.5 metros de distancia el uno del otro, tal y como se observa en la Figura 2.7. En este experimento la distancia es fija y su ejecución en un único escenario es suficiente.





**Figura 2.7** Posición de los nodos sensores en el Experimento C

3. Establecer el nivel de potencia de transmisión deseado y activar el envío de mensajes.
4. Durante las dos primeras horas se deben recolectar medidas del voltaje (utilizando un voltímetro) cada 15 minutos. Después debe realizarse cada 30 minutos. Al mismo tiempo que se mide el nivel de la batería, se debe extraer la información de los nodos sensores con el número de paquetes enviados y recibidos hasta el momento.
5. Este proceso es ejecutado hasta que los dispositivos dejan de enviar o recibir paquetes debido al bajo nivel de sus baterías.
6. Completar las fases 4 y 5 tantas veces como niveles de potencia se quiera testear. En cada iteración las baterías deben ser cargadas o remplazadas por otras con las mismas características.

Después de llevar a cabo el Experimento C, para cada nivel de potencia testado, se obtiene la relación del nivel de la batería con el número de paquetes enviados y paquetes recibidos. Además, se dispone también de un conjunto de medidas que describen el comportamiento de la batería del sensor sin realizar las tareas de envío y recepción de mensajes, lo cual puede ser muy útil para realizar distintas comparaciones.

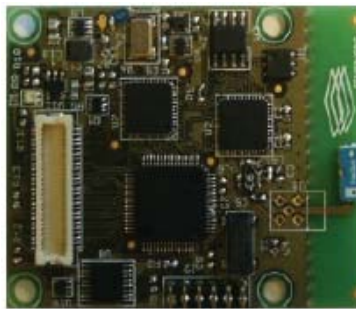
# Capítulo 3

## Caracterización de un nodo comercial

---

### 3.1 Sensor Micro.2420 U100

En el capítulo anterior se ha descrito una técnica mediante la cual se puede realizar la caracterización de cualquier nodo sensor existente en el mercado. Para probarlo, se ha ejecutado su conjunto de experimentos sobre un mote de la compañía *Sensinode*, concretamente uno de sus dispositivos perteneciente a las series Micro, llamado *Micro.2420 U100* [13] (ver su aspecto en la Figura 3.1).



**Figura 3.1** Dispositivo Micro.2420 U100

Este dispositivo contiene un cómodo interfaz para la conexión de distintas unidades sensoras. Además, 2 baterías NiMH se encargan de suministrarle energía. El Micro.2420 U100 integra un flexible microcontrolador TI MSP430 con un radio Chipcon 802.15.4 capaz de transmitir a un alcance de 100 metros, y ofreciendo una tasa de transmisión de datos de 250 kbps. Además puede ofrecer comunicaciones ad hoc con una amplia variedad de topologías. El microcontrolador y la radio son soportados tanto por el sistema operativo FreeRTOS como por TinyOS.

Sus características más importantes son:

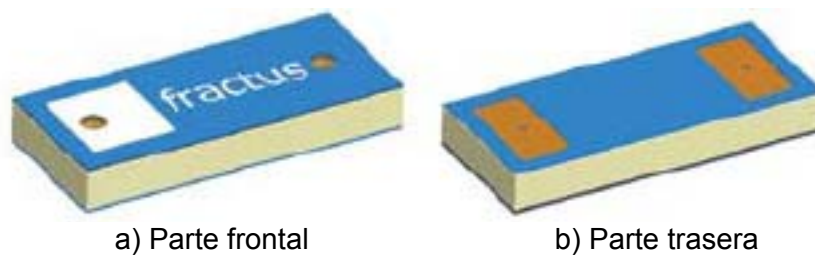
- Potente microcontrolador MSP430 con 10KB de RAM y 256KB de memoria FLASH.

- Múltiple ADCs de 12 bits y dos DAC de 12 bits.
- TI CC2420 802.15.4 compatible con RF-transceiver y tasa de datos de 250 kbps.
- Memoria FLASH de 4Mbit para el almacenamiento de datos.
- Antena instalada en la placa y conector opcional MMCX para situar una antena externa.
- Arquitectura modular, fácilmente expandible añadiendo otros módulos Micro.
- Bajo consumo de potencia, ultra baja potencia en modo sleep y disposición de diversos modos de parado.
- Cada módulo tiene un número ID único.
- Conector externo con 10 IOs digital/analógico
- RoHS compatible
- Tamaño de 40 x 50 mm.
- Operación a 3.3V y un mínimo voltaje en la batería de 1.5V.
- Tiene un módulo para situar 2 baterías NiMH.
- Corriente típica de operación es menor de 25mA.

En los experimentos de la metodología junto a este mote se ha empleado el módulo sensor *A500 Sensor IO* [13], el cual se conecta al Micro.2420 por medio de un cable de 12 pins. Este módulo contiene un sensor de iluminación (Intersil EL7900), otro de temperatura (Microchip TC1047) y un micrófono, todos ellos para llevar a cabo adquisición de datos. Además el módulo contiene dos botones para la interacción del usuario y dos LEDs, los cuales pueden ser controlados mediante software. Los botones son de mucha utilidad en nuestros experimentos ya que por medio de estos se ejecutan acciones como la activación del envío de paquetes o el cambio del nivel

de potencia de transmisión. Todo ello es programado mediante software, al igual que la utilidad de los LEDs, los cuales indican en nuestros experimentos eventos como la llegada de un paquete o el correcto cambio en el nivel de potencia. Por último, señalar que el dispositivo *A500 Sensor IO* tiene una forma cuadrangular de tamaño 40 x 40 mm.

La antena que tiene el nodo sensor incorporada en la placa, es una antena de la compañía *Fractus*, específicamente la antena chip *Fractus Slim Reach Xtend*, la cual ha sido diseñada para trabajar en sistemas inalámbricos que operan en la banda de 2.4 GHz, soportando varios estándares inalámbricos, entre ellos Bluetooth. En realidad esta diseñada para resonar a 2.65 GHz en condiciones de espacio libre, pero al añadir los revestimientos y la influencia de otros factores al trabajar en escenarios reales, provocan una disminución de entre 150 y 250 MHz en la frecuencia de resonancia. La antena posee unas dimensiones de 7 x 3 x 1 mm y puede ser visualizada en la Figura 3.2.



**Figura 3.2** Visualización de la antena chip Fractus Slim Reach Xtend

En el manual de usuario de la antena [26] podemos consultar todas sus especificaciones. La más importante desde el punto de vista de este proyecto es su diagrama de radiación, el cual se muestra en la Figura 3.3.

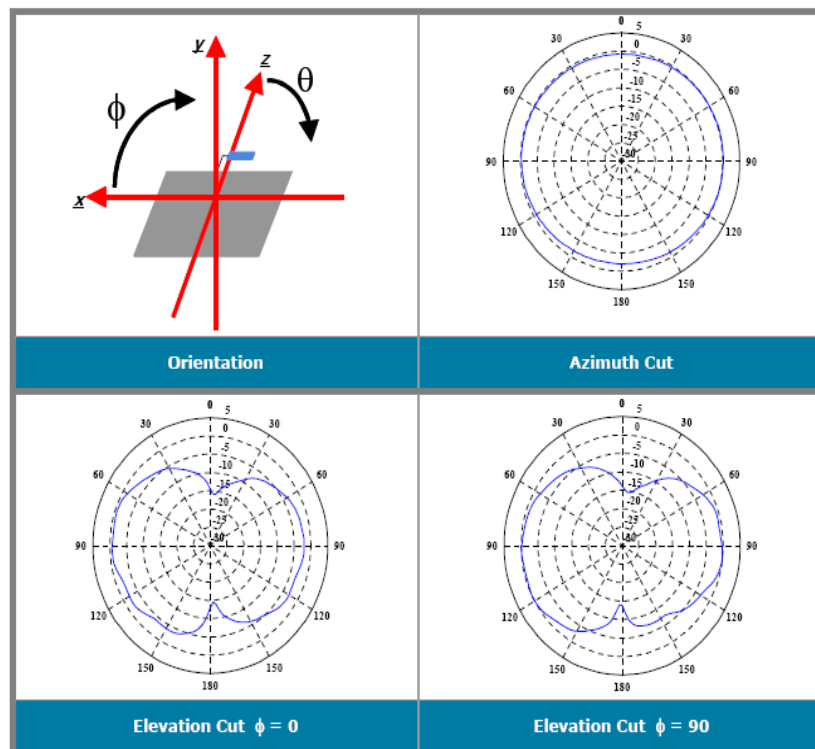
La antena fue testada sobre una placa, cuyas características y especificaciones se pueden apreciar en [26]. Sin embargo, ese estudio no tiene porque reflejar el comportamiento real que esta antena tiene cuando trabaja en el nodo sensor Micro.2420 U100.

La toma de medidas en los tres experimentos de nuestra metodología es llevada a cabo sobre el mismo plano, siendo este plano el paralelo al suelo en cada uno de los entornos donde los dispositivos son testados. Este plano puede apreciarse de forma clara en la Figura 2.7, la cual representa una imagen de los elementos en mitad del

proceso de toma de datos en el Experimento C. Por otro lado, es importante conocer la posición de la antena en el mote testado. El dispositivo de la compañía Fractus puede ser apreciado en la parte derecha del nodo sensor según la Figura 3.1.

Acorde a lo explicado en el párrafo anterior, si se observa el cuadrante superior izquierdo de la Figura 3.3, se aprecia que según está colocada la antena en los ejes, el plano donde se han realizado las medidas se corresponde con el Plano XZ. De modo que para evaluar la máxima directividad de la antena en este plano, nos situamos en él estableciendo  $\Phi = 0$ . Así pues, variando  $\theta$  a lo largo del Plano XZ, obtenemos el diagrama de la parte inferior izquierda de la Figura 3.3. En ese cuadrante se puede ver que la máxima directividad y lóbulo principal de la antena para este plano se encuentra en la dirección negativa del eje X.

En los resultados obtenidos en el Experimento B se intentará comprobar si se verifica esta dirección de máxima directividad cuando la antena Fractus es montada sobre el mote Micro.2420 U100.

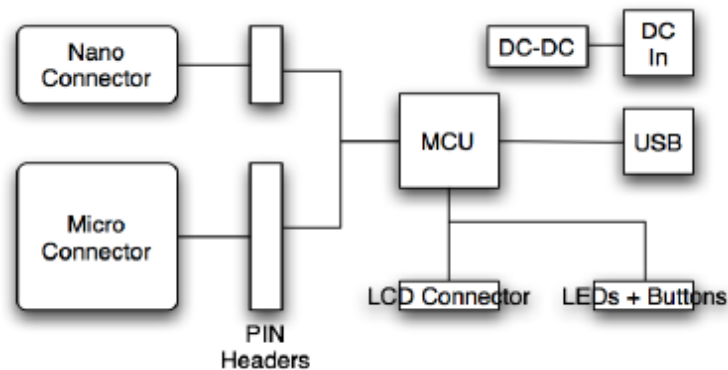


**Figura 3.3** Diagrama de radiación de la antena chip Fractus Slim Reach Xtend

## 3.2 D100 Devboard

Los motes utilizados en los distintos experimentos tenían cargados sus respectivos programas software. Además, en cada uno de esos tests de la metodología existe una fase en la que se realiza la recolección de los datos almacenados en los nodos sensores. Ambos procesos son llevados a cabo gracias al empleo del dispositivo *D100 Devboard* [13].

Esta herramienta posee conectores para interactuar tanto con elementos de las Micro series como de las Nano series, y además puede servir como suministrador de potencia. Su microcontrolador tiene capacidad para programar todos los sensores de ambas series y se conecta al PC mediante un interfaz USB. Además contiene diversos botones y LEDs, así como un conector para interactuar opcionalmente con un display LCD. Su diagrama de bloques se puede apreciar en la Figura 3.4.



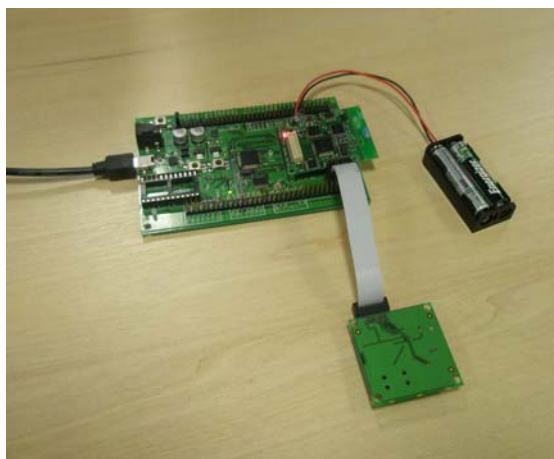
**Figura 3.4** Diagrama de bloques del dispositivo D100 Devboard

Sus características más importantes son:

- Conector USB de alta velocidad a cualquier PC.
- Drivers USB (FTDI 232B chip) disponibles para Linux, SO X y Windows.
- Puede programar todos los módulos de las Micro y Nano series.
- Conector para el display LCD.
- 2 botones y 4 LEDs.

- Conector para ser suministrado de energía externamente.
- RoHS compatible.
- Tamaño de 120 x 80 mm.

Así pues, mediante esta herramienta se programan los motes de *Sensinode* y extraen las medidas llevadas a cabo. Para ello, se conecta el nodo Micro.2420 a la placa de la forma que se aprecia en la Figura 3.5. Se puede observar además del sensor y la placa, el módulo *A500 Sensor IO* unido al mote mediante un cable de 12 pins.



**Figura 3.5** Nodo Micro.2420 junto con el módulo A500 conectados a la placa D100 Devboard.

Para un PC que trabaje en un entorno Linux, las aplicaciones software se cargan en los motes de la siguiente manera: se conecta el nodo a la placa, y ésta a su vez al PC, mediante el interfaz USB. A continuación, ir al directorio donde se encuentra el programa principal que se desea cargar. Finalmente, escribir en consola el comando *make bsl*.

Para la extracción de datos se emplea la herramienta de consola *Minicom*, la cual permite interactuar con el dispositivo de una forma intuitiva y sencilla. Básicamente se interactúa mediante comandos enviados al sensor, los cuales debe ser programado en las aplicaciones software cargadas. Por ejemplo, en nuestro caso se utiliza el lenguaje C y se emplea la instrucción *switch* para interactuar con el mote. Este software es mostrado en secciones posteriores. Por tanto, en nuestros experimentos los datos de

las medidas realizadas fueron mostradas al usuario por consola gracias a la herramienta *Minicom*.

## 3.3 Ejecución del Experimento A

En este apartado se presenta el comportamiento del nodo sensor Micro.2420 U100 en función de la distancia de separación entre los elementos, del nivel de potencia de transmisión y de la localización del experimento. Antes de ello se describe el software cargado en los motes para llevar a cabo este test.

### 3.3.1 Descripción del software

En la descripción del procedimiento completo del Experimento A, se define cual debe ser la función a realizar por los nodos. Se trata de una definición general, de modo que se puede emplear cualquier lenguaje de programación para llevar a cabo la implementación del software. En el Anexo A del presente documento se encuentran las aplicaciones programadas en lenguaje C, las cuales han sido cargadas en el nodo transmisor y el nodo receptor de nuestros tests. El pseudo-código del emisor es el siguiente:

```
1 Inicializaciones
2 Crear Socket UDP
3 for infinito
4 {
5     if botón 1 es pulsado
6     {
7         for cada una de las 3 rondas de envío
8         {
9             for cada uno de los 20 mensajes a enviar
10            {
11                Dormir durante 1 segundo
12                Enviar mensaje
13            }
```



```
14         Dormir durante 5 segundos
15     }
16 }
17 if botón 2 es pulsado
18 {
19     sumar 1 a la variable contador
20     if contador == 4 entonces contador = 0
21     if contador == 0 entonces nivel de transmisión = 100%
22     if contador == 1 entonces nivel de transmisión = 60%
23     if contador == 2 entonces nivel de transmisión = 30%
24     if contador == 3 entonces nivel de transmisión = 10%
25 }
26 }
```

Como se puede apreciar en el pseudo-código, tras las inicializaciones de las variables y la creación del socket, el programa se adentra en un bucle infinito donde la función del sensor está marcada por la pulsación de los dos botones disponibles gracias al empleo del módulo A500. Si se pulsa el botón 1, se activa el envío de mensajes hacia el receptor. Dicho envío está dividido en tres rondas de 20 mensajes cada una, tras las cuales se procede al cambio de posición del nodo sensor en el escenario. Luego se debe pulsar de nuevo este botón para activar otra vez el envío de paquetes. Mediante el botón 2 el usuario podrá modificar el nivel de la potencia de transmisión.

El pseudo-código del receptor es el siguiente:

```
1 Inicializaciones
2 Crear Socket UDP
3 for infinito
4 {
5     if botón 1 es pulsado
6     {
7         Se incrementa en 1 la posición para escribir en los arrays
8     }
9     switch (comando recibido desde teclado)
```

```
10  {
11    'r': se imprimen los arrays con el número de paquetes recibidos
12    's': se imprimen las medidas de la Fuerza de la Señal Recibida
13    'q': se imprimen las medidas del Indicador de Calidad del Enlace
14  }
15  if se reciben datos
16  {
17    Aumento en 1 el número de mensajes recibidos
18    Calculo el valor RSS para el mensaje entrante y lo almaceno
19    Calculo el valor LQI para el mensaje entrante y lo almaceno
20  }
21 }
```

En el receptor, en el bucle infinito se capturan los mensajes procedentes del emisor, almacenándose por cada posición de memoria (en tres arrays diferentes) el número de paquetes recibidos, el RSS y el LQI. Para cada vez que se envíen las tres rondas de paquetes, el usuario debe pulsar el botón 1, con el fin de que los siguientes mensajes recibidos se almacenen en una posición de memoria distinta. Por último, se implementa una estructura *switch*, que nos permite interactuar con el sensor a través de la herramienta *minicom*, con el fin de presentar los datos por pantalla.

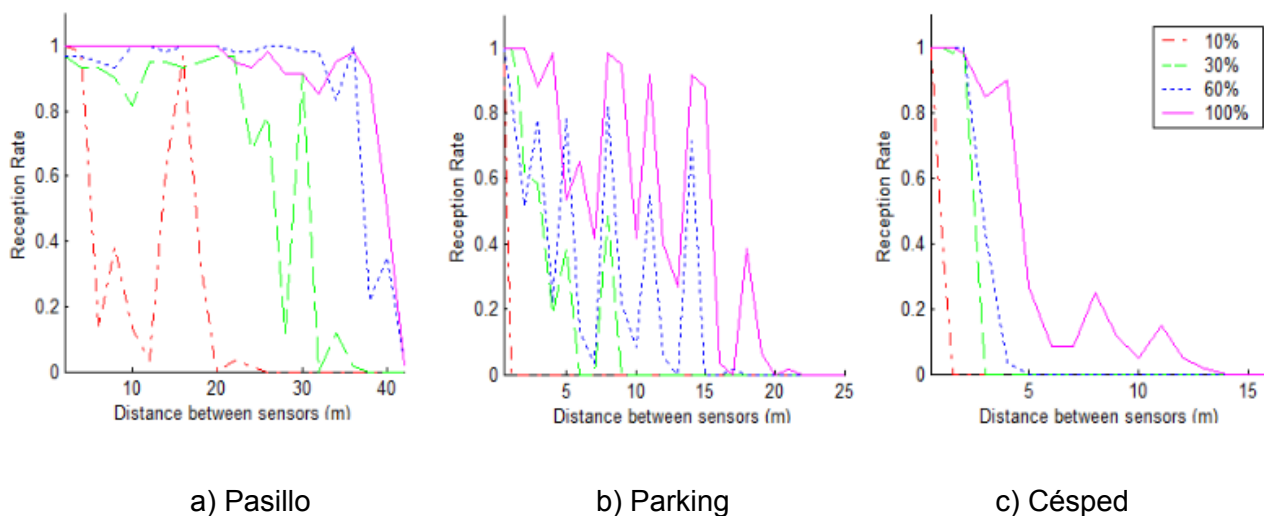
### 3.3.2 Resultados y análisis

A continuación se muestran los resultados obtenidos para cada uno de los parámetros testeados por el Experimento A. Además se realiza un análisis de estos con el objetivo de sacar conclusiones y comparar los hallazgos obtenidos, con las suposiciones realizadas para cada parámetro en el Capítulo 2.

#### ■ Tasa de recepción de paquetes.

La Figura 3.6 muestra la relación de la tasa de paquetes recibidos por los nodos con la distancia y el nivel de potencia de transmisión. Señalar que este nivel está expresado en porcentaje sobre la potencia máxima que puede emitir el nodo, ya que al usuario no le interesa conocer el valor específico, debido al hecho de que cualquier otra familia de nodos podría emitir a potencias muy superiores o muy inferiores, siendo por tanto más

correcto realizar el análisis según valores porcentuales de este parámetro, siendo el 100% la máxima potencia que el dispositivo podría transmitir.



**Figura 3.6** Relación entre la tasa de recepción, la distancia y la potencia de emisión para cada entorno.

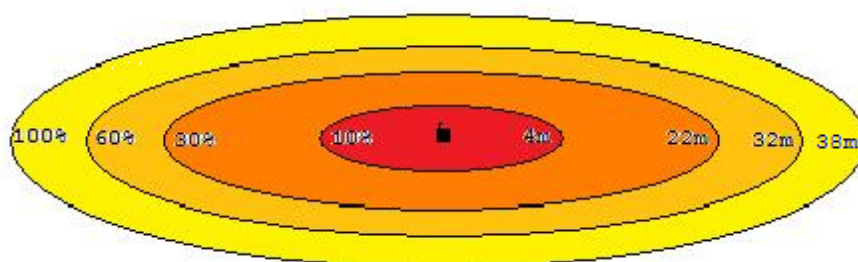
Observando la figura se aprecia que al aumentarse la distancia de separación entre los elementos, en general la tasa de recepción de paquetes disminuye. Además cuando el nivel de la potencia de transmisión aumenta, la tasa de paquetes procesador por el nodo receptor se incrementa, al igual que lo hace el alcance máximo de la comunicación (máxima distancia a la que se reciben mensajes). Todas estas tendencias ocurren para cada una de las distintas localizaciones donde el Experimento A fue ejecutado. De este modo se cumplen las suposiciones iniciales que listábamos en la subsección 2.3.1.

Por otro lado, se puede apreciar en la figura que para la probabilidad de recepción de paquetes los resultados obtenidos son bastante pobres, ya que el alcance en la comunicación es muy corto para las dos localizaciones de exterior. Solamente en el pasillo se obtienen valores decentes en cuanto al alcance, siempre y cuando se empleen valores elevados de la potencia (con el 60% no se obtienen malas prestaciones). Sin embargo, desde el punto de vista energético, en una WSN se debería emplear potencias lo más bajas posible. La utilización o no de este tipo de motes para crear una red de sensores depende de las propias características y prestaciones del sistema.

Cuando un sistema de telecomunicación es creado, se debe realizar para que se cumplan ciertos criterios de calidad. La probabilidad o tasa de paquetes recibidos es

uno de ellos (en ocasiones lo es su complemento: la tasa de paquetes perdidos). Normalmente, se suele diseñar el sistema inalámbrico para que todos los elementos de la comunicación se comporten de forma que esta probabilidad de entrega de paquetes se mantenga por encima de un umbral. Según el escenario, este criterio se podría conseguir fácilmente si todos los nodos transmiten utilizando la máxima potencia. Sin embargo, esto no es posible en una WSN, donde la optimización del mínimo consumo energético prevalece sobre el resto de características. Así pues, es responsabilidad del diseñador de la red de sensores, el llevar a cabo una toma de decisiones apropiada con el fin de que la WSN trabaje correctamente y por un largo periodo de tiempo.

Siguiendo el hilo temático de lo explicado en el párrafo anterior, se describe el siguiente ejemplo: se desea desplegar una WSN en el interior de un edificio, demandándose el mayor ahorro energético posible y una probabilidad de recepción mínima de 0.9 para cada uno de los elementos que formarán la red. El problema será resuelto empleando los datos del Experimento A y por tanto observando la gráfica del pasillo de la Figura 3.6-a. Así, para cada nivel de la potencia se escoge el alcance máximo mostrado en la gráfica para el cual la tasa de recepción es 0.9 o superior, de modo que los nodos usen dicha potencia si el sensor al que van a enviar información está a esa distancia o más cerca. En caso de que se encuentren más alejados emplearía una potencia superior. De este modo, los datos obtenidos son: 4, 22, 32 y 38 metros de alcance para el 10%, 30%, 60% y 100% de la potencia, respectivamente. Para este ejemplo, se puede apreciar en la Figura 3.7 la potencia a la que debe transmitir cada nodo acorde a la distancia del sensor destino, cumpliendo siempre con los criterios de mínimo consumo y de tasa de recepción igual o superior a 0.9.



**Figura 3.7** Mapa de configuración del nodo sensor para optimizar la energía y la tasa de recepción.

Una de las cuestiones que se dejó abierta en la sección 2.3.1 fue la comprobación de la proporcionalidad entre el alcance máximo y el nivel de la potencia transmitida. Gracias a los datos obtenidos en el ejemplo anterior se puede afirmar, que al menos

para el conjunto de medidas realizadas en este proyecto, estos parámetros no son directamente proporcionales.

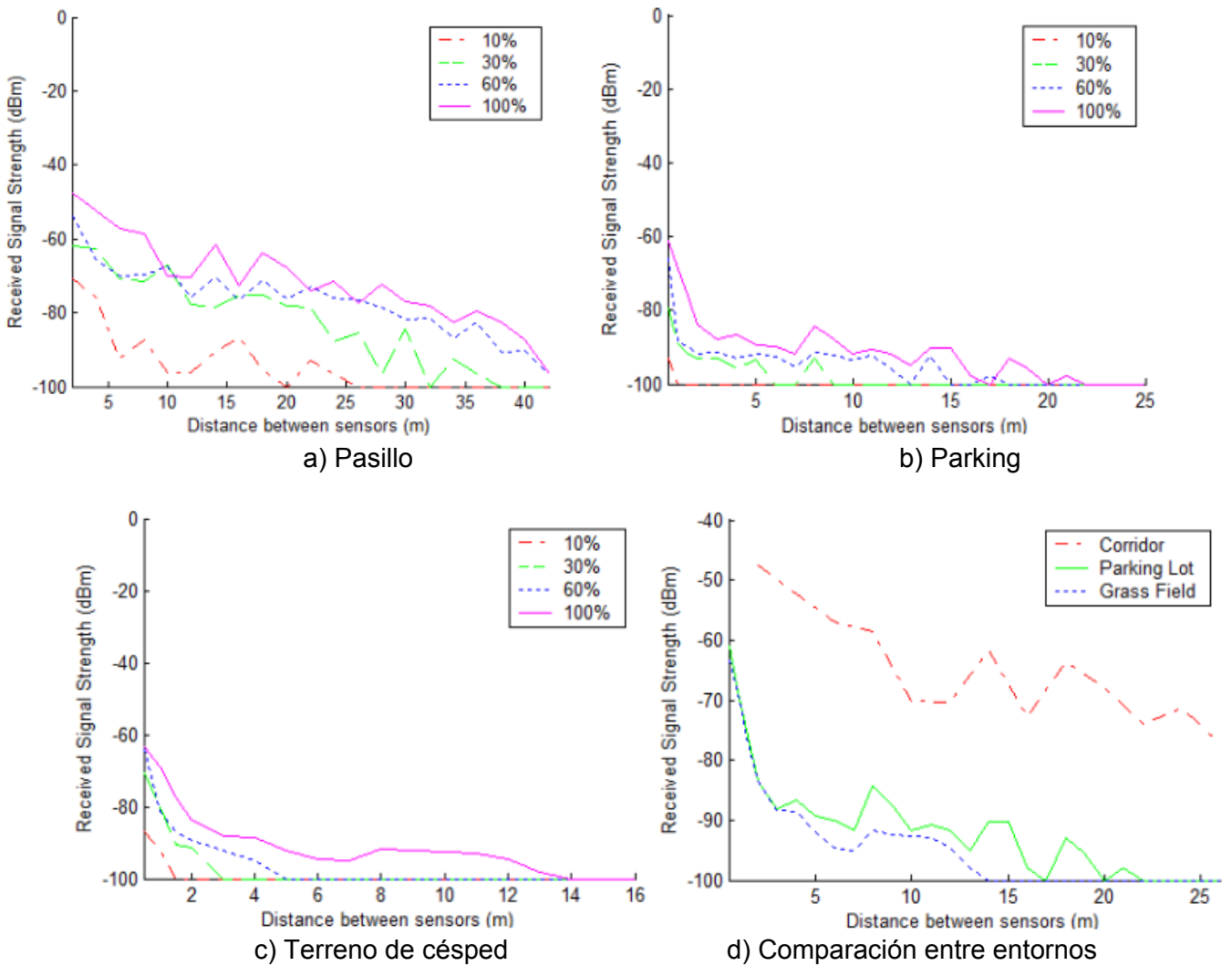
En la Figura 3.6 se puede observar la existencia de zonas donde la comunicación presenta peores prestaciones que en otras áreas, estando situado el nodo receptor en una posición más cercana al emisor, es decir, existen distancias donde la tasa de recepción es significativamente menor que en otras posiciones más alejadas. El fenómeno multi-camino podría ser una de las causas de la existencia de estas irregularidades en la comunicación. Este fenómeno ocurre cuando la antena receptora captura la misma señal procedente de dos o más rutas distintas, coincidiendo los rayos en fase en recepción. Su influencia se apreciaría principalmente en el entorno del parking de automóviles (ver la Figura 3.6-b), donde los rayos reflejados pueden ser resultado de su choque en objetos como árboles o coches. Prueba de ello puede ser la no aparición de estas irregularidades en el terreno de césped debido a que, al contrario que en el parking, en el momento de ejecución del experimento no había ningún posible elemento que pudiera ocasionar el reflejo de la señal.

Como fácilmente se aprecia en la Figura 3.6, el máximo alcance en la comunicación se obtiene en el pasillo. Este hecho podría tener también relación con el multi-camino, debido a que al ser tan estrecho el pasillo, las señales rebotarían con una mayor facilidad en las paredes, llegando al receptor un mayor número de rayos y por tanto incrementándose la potencia de la señal recibida, hecho que será comprobado en la siguiente subsección.

De modo que al igual que en los estudios descritos en [2] y [23], se obtienen resultados que presentan ciertas irregularidades en la comunicación entre nodos sensores de una WSN.

#### ■ **Fuerza de la Señal Recibida.**

Las gráficas obtenidas de las medidas tomadas en el Experimento A, respecto a la Fuerza de la Señal Recibida, pueden ser observadas en la Figura 3.8. Señalar que el mínimo valor del RSS obtenido en el experimento fue -94 dBm. Sin embargo, en las curvas se pueden apreciar valores menores a este. Estos valores no deben crear confusión, ya que se corresponden a 0 mW, pero se ha representado de este modo con el fin de dar más claridad a las gráficas.

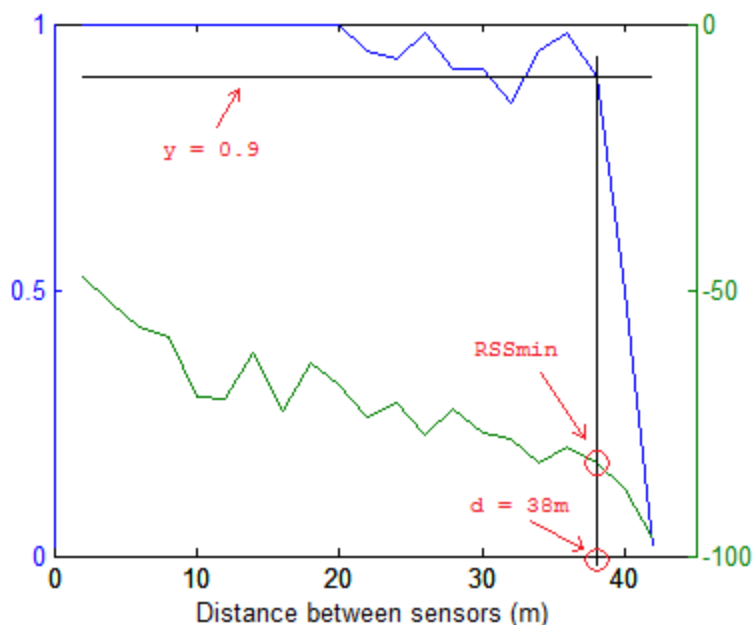


**Figura 3.8** Relación entre el RSS, la distancia y la potencia de transmisión para cada entorno.

Observando la Figura 3.8 se puede afirmar que para cada una de las localizaciones la Fuerza de la Señal Recibida por el nodo decrece a la vez que la distancia de separación es incrementada. Además, lógicamente para una distancia fija dicha fuerza recibida se hace más intensa cuando se aumenta la potencia de transmisión. De este modo se verifican las suposiciones realizadas por diversos autores en sus trabajos en [1] y [24], donde se definían esas relaciones y además se incidía que la función de RSS respecto a la distancia decaía de forma más abrupta en los primeros metros, lo cual se cumple también para nuestra investigación tal y como se ve en la figura.

Además, se puede ver la existencia de ciertas irregularidades en la comunicación, principalmente en las gráficas del pasillo y del parking. Al igual que en los resultados

de la tasa de recepción, el multi-camino puede ser el fenómeno que provoque su aparición. Respecto al parámetro RSS, en la Figura 3.8-d puede apreciarse la gran diferencia existente entre el entorno de interior y las localizaciones exteriores, de las cuales el terreno de césped presenta las peores características. En esa gráfica las curvas de la fuerza recibida son representadas para los sensores configurados para transmitir al 100% de la potencia.



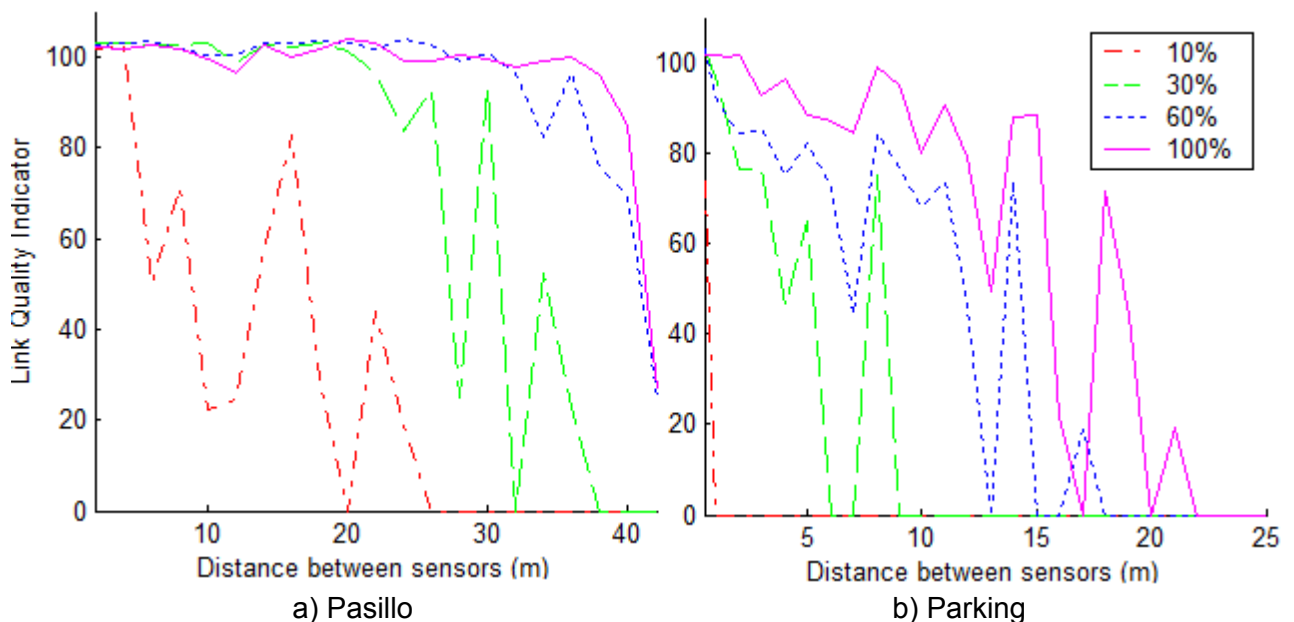
**Figura 3.9** Obtención del RSS mínimo

Por otro lado, al diseñarse una WSN, podría ser interesante conocer el valor de la Fuerza de la Señal Recibida mínima ( $RSS_{\min}$ ) por un nodo, a partir del cual se asegura que la probabilidad de paquetes recibidos es igual o superior a un determinado umbral. De este modo, el diseñador sólo tendría que colocar y configurar todos los sensores de la red para que reciban un valor de fuerza  $RSS_{\min}$  o superior, asegurándose de este modo un correcto comportamiento de la red respecto a la recepción de paquetes. Viéndolo con un ejemplo: se considera un valor mínimo para la probabilidad de recibir paquetes de 0.9. Para un nivel de potencia del 100%, observando la gráfica de las medidas para el pasillo en la Figura 3.9 (donde se representa el RSS en el eje y de la derecha y la tasa de recepción en el eje y de la izquierda), se obtiene la intersección de la recta  $y = 0.9$  con la función de la probabilidad de recepción. Entonces se traza una perpendicular que corte la función de RSS, obteniendo así el valor mínimo necesario ( $RSS_{\min} = -83$  dBm) y también la distancia a la cual se obtiene ese valor (38 metros).

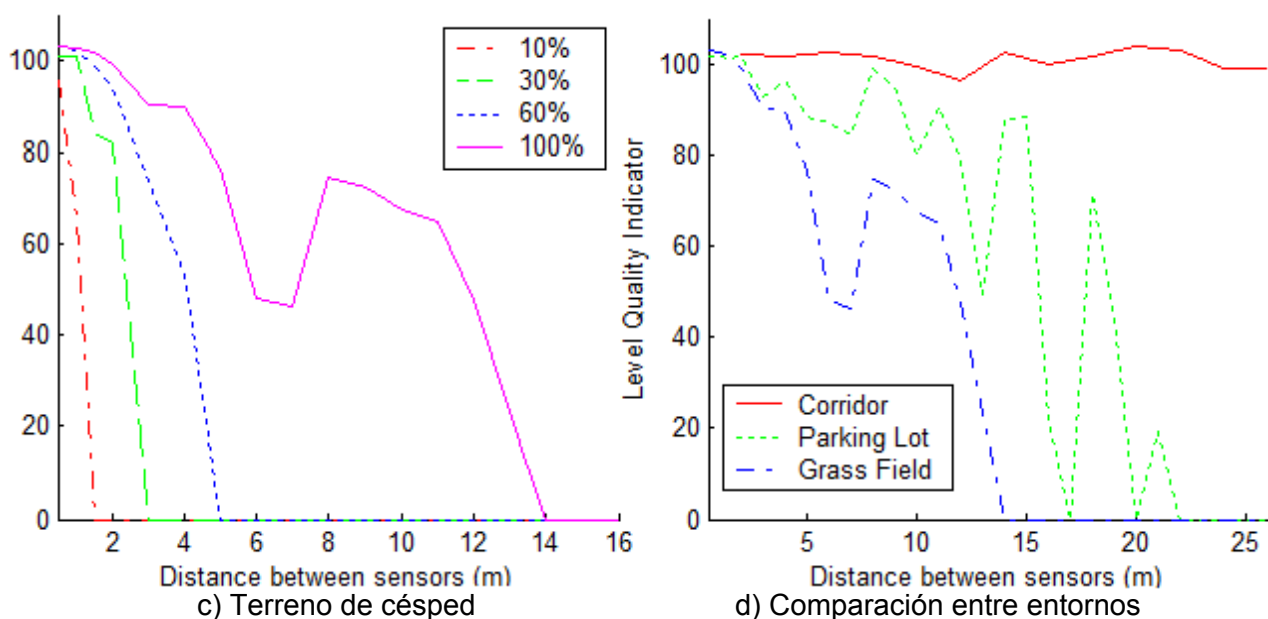
■ **Indicador de Calidad del Enlace.**

En la Figura 3.10 se muestran los resultados obtenidos del parámetro que calcula el error de los paquetes recibidos con el fin dar una medida de la calidad del enlace. Su función está representada con relación a la distancia y para distintos valores de la potencia de transmisión. Se puede observar que cuando la separación entre dispositivos crece, el enlace presenta peor calidad. Así, para valores más altos de la potencia se obtiene un incremento del parámetro LQI y por tanto una disminución del error. Estas dos relaciones se cumplen de forma general para cada una de las localizaciones, existiendo numerosas irregularidades principalmente en pasillo y en el parking, al igual que ocurría con los otros dos parámetros testeados mediante el Experimento A. Al igual que con esos parámetros, el fenómeno multi-camino es supuestamente el origen de dichas irregularidades. Por último, apreciar en la Figura 3.10-d, que el enlace del pasillo se corresponde con el que posee mayor calidad, mientras que el que presenta peores prestaciones es el enlace del terreno con vegetación. Las gráficas mostradas en esa figura se corresponden con medidas tomadas para un nivel de la potencia de transmisión del 100%.

Por tanto a la vista de estos resultados, si se desea diseñar una red de sensores donde es característica primaria que la información llegue al destino sufriendo la mínima alteración posible, es decir, sin que los paquetes dispongan de errores, se deberá utilizar una localización para su despliegue que presente un comportamiento como el entorno del pasillo en este experimento, y nunca como el de los otros dos escenarios.







**Figura 3.10** Relación entre el LQI, la distancia y la potencia de transmisión para cada entorno.

## 3.4 Ejecución del Experimento B

La respuesta del dispositivo Micro.2420 U100 en función de su ángulo de orientación es estudiada en este apartado. El test ha sido ejecutado utilizando dos de estos nodos sensores (un emisor y un receptor) en el pasillo y el parking, y cinco (un emisor y cuatro receptores) en el terreno con vegetación.

### 3.4.1 Descripción del software

Las aplicaciones empleadas en el Experimento B son exactamente las mismas a las utilizadas en el A, las cuales son descritas en la subsección 3.3.1.

### 3.4.2 Resultados y análisis

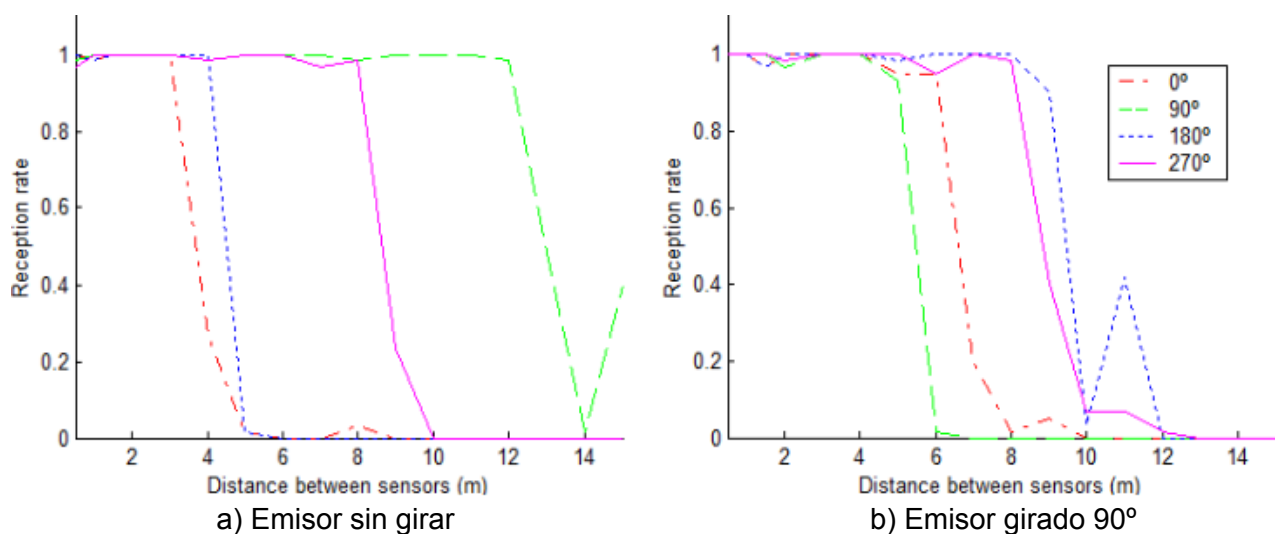
La orientación de los nodos es el único parámetro testado en el Experimento B. Los resultados obtenidos y su análisis se presentan a continuación.

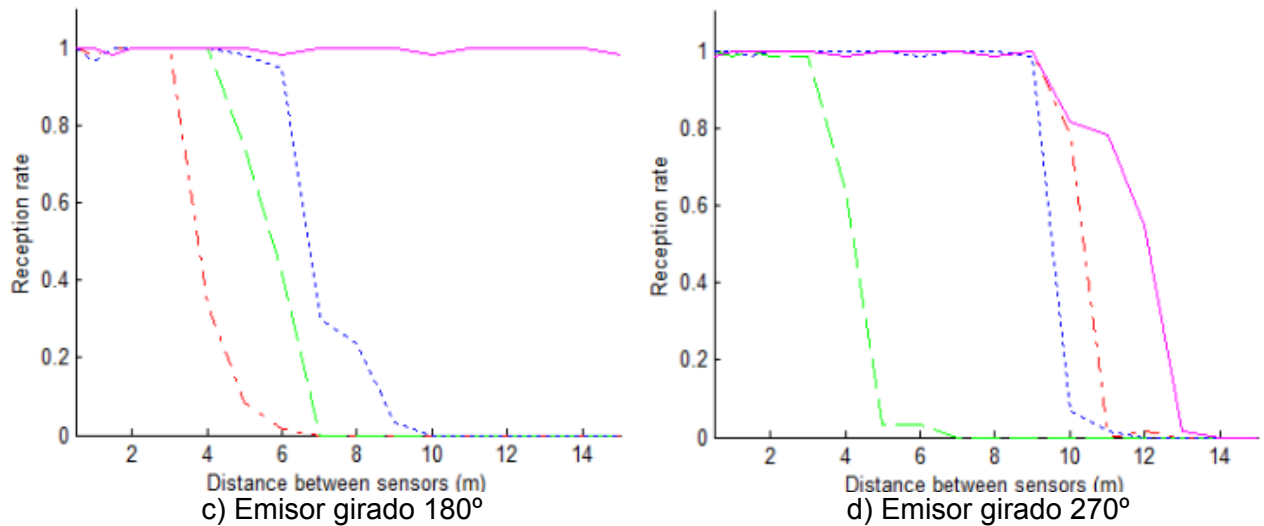
■ **Orientación de los nodos.**

Se han testeado 16 posiciones diferentes entre un nodo emisor y un nodo receptor (cuatro en el caso del césped) para las tres localizaciones. Además para cada una de esas posiciones se han recolectado datos de la tasa de recepción, la Fuerza de la Señal Recibida y el Indicador de Calidad del Enlace. Todos estos recogidos en función de la distancia de separación entre dispositivos y para una potencia de transmisión fija. En total disponemos de cientos de mediciones y un gran número de gráficas.

Por otro lado, el objetivo fundamental del Experimento B es caracterizar los nodos respecto a su orientación óptima, es decir, analizar los resultados obtenidos de forma que se identifique cual es la mejor posición de los sensores con el fin de que se incrementen lo máximo posible las prestaciones del sistema. Para llevar a cabo dicha identificación, es suficiente con mostrar el estudio referente a la tasa de recepción de mensajes según los distintos ángulos de orientación, para la localización del terreno de césped.

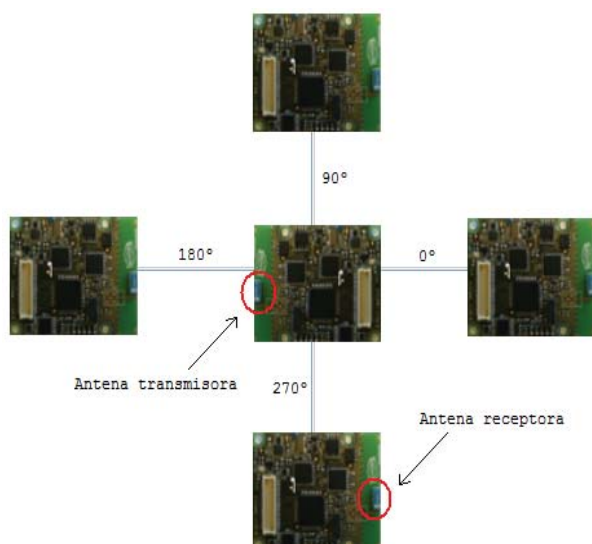
Así pues, la Figura 3.11 muestra los resultados obtenidos en el terreno de césped cuando el experimento es ejecutado completamente con un nivel de potencia en la transmisión de paquetes del 100%. Cada una de las cuatro gráficas que pueden observarse en la figura se corresponden a las cuatro diferentes posiciones de giro del nodo transmisor ( $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  y  $270^\circ$ ). Además, cada gráfica representa la función de la probabilidad de recepción para los cuatro receptores, cada uno situado en un ángulo y posición distinta respecto al nodo emisor.





**Figura 3.11** Relación entre la tasa de recepción y la distancia para cada posición del emisor.

Como ya se ha indicado, el objetivo es encontrar la posición concreta del emisor y del receptor que optimice las prestaciones en la comunicación, o para este caso que maximice el número de mensajes recibidos. Apreciando la Figura 3.11, se puede ver que la mejor respuesta en la comunicación se obtiene cuando el emisor está girado  $180^\circ$  sobre su eje, y se comunica con el receptor situado en la posición a  $270^\circ$  respecto a él (ver la Figura 3.11-c). En ese caso la antena emisora esta transmitiendo con su máxima directividad hacia el receptor, el cual captura la información orientado con su mayor directividad también. Puede verse más fácil en el esquema de la Figura 3.12, donde se representa la configuración del experimento a partir del cual se recolectaron las medidas representadas en la Figura 3.11-c. Se puede apreciar el emisor en el centro girado sobre si mismo  $180^\circ$ , y el receptor que obtiene la más alta tasa de recepción situado en la posición de  $270^\circ$  respecto a la fuente de paquetes. Es este escenario, los lóbulos con mayor ganancia de ambas antenas están enfrentados.



**Figura 3.12** Esquema de la comunicación entre nodos en el Experimento B.

Para el esquema de la Figura 3.12, observando el diagrama de radiación se aprecia que la antena transmisora presenta dos ceros, uno en la dirección del receptor situado en  $0^\circ$  y otro en la dirección del situado a  $180^\circ$ . El primer cero se puede apreciar fácilmente en la gráfica de la Figura 3.11-c, ya que en ese enlace de la comunicación es donde se obtiene el menor alcance.

La orientación del nodo sensor con máxima directividad se corresponde con el lóbulo de mayor ganancia mostrado en el diagrama de radiación teórico de la antena *Fractus*, el cual se representa en la Figura 3.3. Por tanto, se ha demostrado que los nodos presentan las mejores características para la comunicación en cuanto a la orientación para la posición que se presumía.

### 3.5 Ejecución del Experimento C

Para la ejecución de este experimento, donde se testea la eficiencia energética de los nodos, se utilizaron baterías AA NiMH de 1.5V de la marca *Energizer*. En el test inicialmente se realizaba un pequeño experimento donde se recogían datos para un nodo sensor incomunicado. Posteriormente se ejecutaba el experimento principal con un sensor emisor de paquetes y otro receptor. Todo ello se ha realizado solamente en el entorno del pasillo.

### 3.5.1 Descripción del software

En este caso se han creado tres programas diferentes, uno para el nodo sensor incomunicado y otros dos para los motes transmisor y receptor. Para más detalles sobre el código se debe ver el Anexo B. El pseudo-código del dispositivo que envía mensajes es el siguiente:

```
1 Inicializaciones
2 Crear Socket UDP
3 for infinito
4 {
5     if botón 1 es pulsado
6     {
7         if variable flag == 0 entonces flag =1
8         else flag = 0
9     }
10    if flag == 1
11    {
12        for cada uno de los 10 mensajes a enviar
13        {
14            Dormir
15            Enviar mensaje
16            Aumentar en uno la variable que almacena el número de mensajes enviados
17        }
18    }
19    switch (comando recibido desde teclado)
20    {
21        's': se imprime la variable con el número de paquetes enviados
22        '+': se aumenta el nivel de la potencia de transmisión en un 10%
23        '-': se disminuye el nivel de la potencia de transmisión en un 10%
24    }
25 }
```

Destacar de este programa, que mediante la pulsación del botón 1, se activa el envío de mensajes, el cual no se interrumpirá hasta que se pulse de nuevo este botón. Este

control se lleva a cabo mediante el uso de la variable *flag*, la cual hace la función de semáforo.

El pseudo-código del receptor es el siguiente:

```
1 Inicializaciones
2 Crear Socket UDP
3 for infinito
4 {
5     if se recibe el caracter 'r' desde teclado entonces se muestra el número de mensajes recibidos
6     if se recibe un paquete se aumenta en uno el número de mensajes recibidos
7 }
```

Por otro lado, el pseudo-código perteneciente al nodo incomunicado es:

```
1 Inicializaciones
2 Crear Socket UDP
3 for infinito
4 {
5     if botón 1 es pulsado
6     {
7         if variable flag == 0 entonces flag =1
8         else flag = 0
9     }
10    if flag == 1
11    {
12        for cada uno de los 10 mensajes a enviar (aunque en realidad no se envían)
13        {
14            Dormir
15            Aumentar en uno la variable que almacena el número de mensajes enviados
16        }
17    }
18 }
```

El programa de este experimento inicial contiene la misma carga operativa que la aplicación cargada en el nodo emisor, excepto la ejecución de la propia sentencia que realiza el envío de la información vía socket (*socket\_sendto*). El usuario simplemente debe pulsar el botón 1 al comienzo del experimento, para que el sensor se introduzca en el proceso de envío (pero sin transmitir paquetes) y permanezca ahí hasta que la batería se agote.

### 3.5.2 Resultados y análisis

La vida de la batería de los nodos es el único parámetro testeado en el Experimento C. Las medidas han sido tomadas para dos niveles distintos de la potencia: el 10% y el 100%. Los resultados obtenidos y su análisis se presentan a continuación.

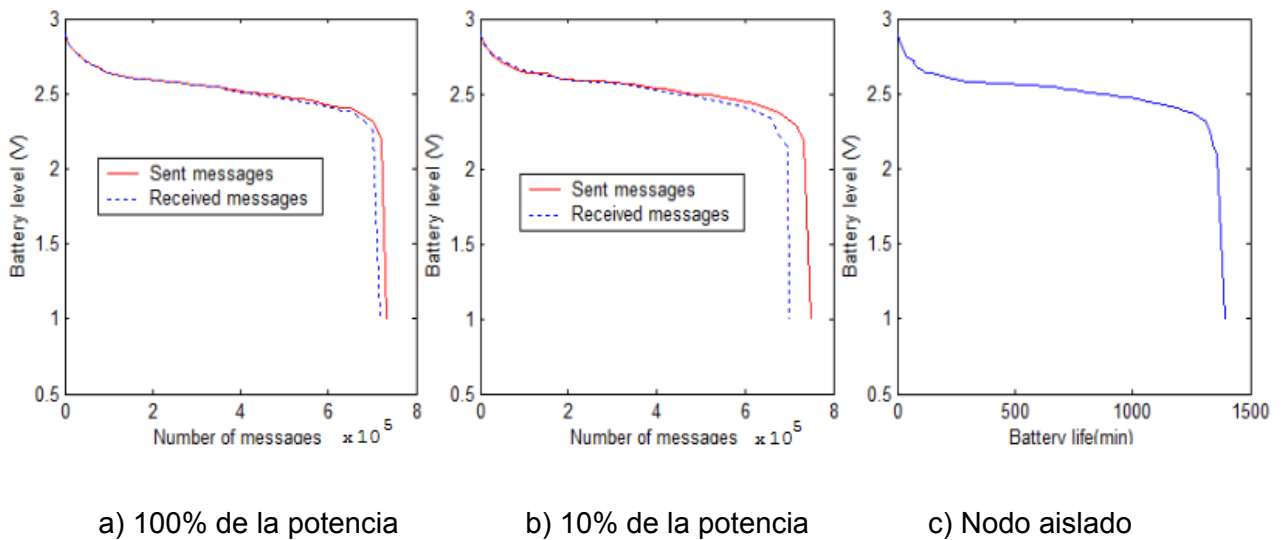
#### ■ Vida de la batería.

La Figura 3.13 muestra por un lado (gráficas a) y b)) la relación entre el nivel de la batería de los sensores y el número de mensajes enviados y recibidos, para dos niveles de potencia distintos, 10% y 100%. Por otro lado, la gráfica c) representa la vida de la batería en función de su nivel para un nodo que no se comunica. Como relación directa se puede apreciar que para dos sensores con idéntico nivel y tipo de batería, la energía se agota antes en el dispositivo receptor que en el emisor. Es decir, para el software cargado en los dispositivos, el proceso realizado por el receptor consume más energía que el proceso que realiza el emisor. La explicación se encuentra en que el transmisor, entre mensaje y mensaje enviado, se encuentra durmiendo (o en modo *sleep*), mientras que el receptor tiene a su CPU trabajando ya que se encuentra en un bucle infinito a la espera de recibir paquetes. Aquí demostramos la importancia de enviar a dormir los nodos, y su consecuencia respecto al consumo energético.

Por otro lado, en la Figura 3.13 se puede apreciar que el número de mensajes enviados utilizando un nivel del 100% de la potencia es menor que al emplear el 10%. Esto ocurre debido a que el nivel de la batería del emisor se agotó antes en el nodo configurado para transmitir al 100% y por tanto se enviaron menos paquetes. Este hecho demuestra que a mayor potencia de transmisión en los nodos, mayor consumo energético, tal como se presumía en el Capítulo 2 del proyecto. Se debe apreciar

también que el número de paquetes recibidos para ambos niveles de potencia es prácticamente el mismo. Este es uno de los hechos que confirman la correcta ejecución del experimento, ya que el factor potencia de emisión evidentemente no debe afectar al nodo receptor.

La gráfica c) muestra la relación entre la vida de la batería y su nivel de voltaje para un nodo que no transmite ni recibe paquetes, y que se encuentra simplemente operando del mismo modo que lo hace el nodo emisor. El dato más significativo es la baja vida de la batería del sensor en este experimento (aproximadamente 23 horas), en comparación con el tiempo que debería durar la energía en un dispositivo que va a formar parte de una WSN (en algunos motes su batería perdura durante años). La explicación es la gran carga operativa a la que se ve expuesta la CPU de este sensor comparado con la que realizaría otro nodo trabajando en una WSN real. Mientras que el dispositivo en nuestro experimento se mantiene realizando operaciones una gran cantidad de tiempo, un nodo normal en la mayoría de las redes reales se encuentra en modo *sleep* casi el 100% del tiempo. Sin embargo, se tuvo que realizar el diseño de nuestro experimento de esta configuración, ya que en otro caso las medidas hubiesen sido de una duración incalculable.



**Figura 3.13** Funciones del nivel de batería en relación a los mensajes enviados/recibidos y al tiempo.

Otro aspecto reseñable es que la vida de la batería en el sensor aislado es más larga que la del emisor y receptor en ambos experimentos de la potencia. Este aspecto es lógico, ya que dicho sensor incommunicado realiza las mismas operaciones que el nodo



transmisor, a excepción de las instrucciones que ejecutan el propio envío de la información. Sin embargo, esa diferencia temporal no es muy abrupta, existiendo aproximadamente sólo 100 minutos de diferencia entre la vida de la batería del sensor aislado y la vida del emisor que transmite con el 10% de la potencia, a pesar de que este sensor se encuentre enviando 10 paquetes por segundo sin interrupción. Por tanto, esto significa que según los resultados de los que disponemos, para nuestros experimentos podemos afirmar que los procesos de envío y recepción de mensajes no son factores críticos a la hora de estudiar la eficiencia energética en los nodos de una WSN, teniendo un carácter mucho más importante aspectos como el procesado de código realizado por la CPU o la frecuencia con la que el dispositivo se encuentra en modo *sleep*. Explicado de otro modo, en nuestros tests las tareas de envío o recepción de paquetes consumen una energía despreciable con respecto a la carga operativa de la CPU del nodo sensor. Sin embargo, la existencia de multitud de artículos donde se afirma que el consumo producido por las comunicaciones es el mayor responsable del gasto energético, nos hace replantearnos la veracidad de nuestros hallazgos. Como explicación a esto, podría existir algún error en los programas cargados en los dispositivos, algún fallo en su hardware o incluso en el proceso experimental.

Finalmente señalar que las todas curvas representadas en la Figura 3.13 tienen un comportamiento similar a la función referencia para las baterías NiMH mostrada en la Figura 2.6 de la sección 2.5.1. Todas ellas poseen caídas de voltaje con una alta pendiente al comienzo y al final de la función, y un descenso suave del nivel de batería durante casi todo el tiempo de su ciclo de vida. El valor de voltaje a partir del cual se produce la caída final de la curva es bastante similar en cada experimento, siendo este 2.37 V aproximadamente.

# Capítulo 4

## Conclusiones

---

El deseo por la monitorización de multitud de diferentes entornos y fenómenos ha provocado que las Redes de Sensores Inalámbricas constituyan una de las tecnologías que han adquirido mayor importancia en los últimos años. Como consecuencia, durante este tiempo han aparecido numerosos trabajos para la investigación y el desarrollo de WSN. El presente proyecto representa una aportación más, mediante la definición de una metodología, la cual permite la caracterización de nodos en este tipo de redes en diferentes localizaciones.

El diseñador interesado en el despliegue de una red de sensores, puede seleccionar del mercado entre una gran gama de distintos motes, cada uno con unas características muy diferentes el uno del otro. Sin embargo, se encuentra con el problema de que el comportamiento de estos dispositivos no está lo suficientemente detallado, en sus hojas de características, como para predecir como se comportarán en el escenario real en el que van a trabajar. De ahí la necesidad de la aparición de un trabajo como el nuestro, que permita al creador de la red testear de antemano las características de los nodos, antes de que el sistema sea desplegado completamente.

Nuestra técnica de testeo se divide en tres experimentos principales. En el primero de ellos se estudia el comportamiento de los nodos para los parámetros de la tasa de recepción de paquetes, la Fuerza de la Señal Recibida y el Indicador de Calidad del Enlace. En el siguiente experimento, se analizan las características de los motes según su ángulo de orientación en el espacio, estudiándose finalmente en el último, la respuesta energética de los sensores inalámbricos.

Tras la definición de la metodología, en el Capítulo 3 se presentan los resultados y el análisis de la ejecución de nuestra técnica sobre un mote cualquiera del mercado, específicamente el Micro.2420 U100 de la compañía *Sensinode*. Muchos de los hallazgos descritos en este trabajo corroboran las afirmaciones relatadas en los estudios de otros autores. Así por ejemplo, se demuestra que las prestaciones en la comunicación de los nodos, decrecen al aumentarse la distancia de separación entre ellos y mejoran al incrementarse el nivel de la potencia de transmisión. Por otro lado,

se indica que el fenómeno multi-camino puede ser el causante directo de la aparición de numerosas irregularidades en la comunicación. Mediante la ejecución del Experimento B se comprueba que el comportamiento llevado a cabo por los nodos sensores en función de su ángulo de orientación, en un entorno real, siguen el comportamiento descrito de forma teórica por el diagrama de radiación de su antena. Por último, del experimento sobre la vida de las baterías de los sensores, el aspecto más importante a destacar es que, acorde a nuestros resultados sobre el Micro.2420 U100, el uso intensivo de otros subsistemas ajenos a la radio (en especial la CPU) eleva el consumo energético, demostrando esto la importancia de utilizar el modo *sleep* en las aplicaciones.

De todos modos, aunque se haya realizado un minucioso análisis de los resultados obtenidos tras la ejecución de la metodología, es crucial reafirmar el objetivo fundamental de este documento, que no es otro que la descripción de un proceso sistemático experimental que permita el testeo de cualquier nodo comercial, para su uso en el despliegue de una WSN real, con el fin de que las prestaciones y características del sistema sean optimizadas.

# Anexo A

## Código de los Experimentos A y B

---

### A.1 Nodo Emisor

```
/* Standard includes. */
#include <stdlib.h>
#include <signal.h>
#include <string.h>

/* Scheduler includes. */
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"

#include "bus.h"
#include "gpio.h"

#include "debug.h"

static void vapplitest( void *pvParameters );
#include "socket.h"
#include "rf.h"
#include "control_message.h"
#include "neighbor_routing_table.h"

/* Message types */
#define SINK_REQUEST 0x77
#define SENSOR_DATA 0xaa

/* Control types */
#define LED_CONTROL 0xe0
#define NORMAL_DATA 0xe1
/* Control attributes */
#define BUTTOM_1 0x10
#define BUTTOM_2 0x20

#define NUM_MSG 20
#define NUM Rondas 3

extern sockaddr_t mac_long;

sockaddr_t discover_address =
{
    ADDR_802_15_4_PAN_LONG,
    { 0xFF, 0xFF, 0xFF, 0xFF,
      0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF },
    61619
};

sockaddr_t data_address =
{
```

```
ADDR_BROADCAST,
{ 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
61619
};

int main( void )
{
    LED_INIT();
    if (bus_init() == pdFALSE)
    {
    }

    xTaskCreate( vapplitest, "Term", 250, NULL, ( tskIDLE_PRIORITY + 1 ), NULL );
    vTaskStartScheduler();
    return 0;
}
/*-----*/
typedef uint8_t b1w_reg[8];
buffer_t *buffer;
buffer_t *data;
socket_t *discover_socket=0, *control=0, *data_rx=0;
control_message_t *msg;
int8_t ping_active=0;
portTickType ping_start;

static void vapplitest( void *pvParameters )
{
    uint8_t i=0, sink_active=0, type=0, msg=0;
    uint8_t length, set_led;
    uint16_t discover_counter = 0, count=100;
    portTickType xLastWakeTime;
    pause(200);
    debug_init(115200);
    pause(300);

    uint8_t count_power=0, flag =0, sem=0, pos=0, exper=0;
    uint8_t tx_power=0;
    int16_t byte;

    int8_t j=0;

    int8_t plr[45];
    int16_t ssrc[45];
    int16_t lqi[45];

    vTaskDelay(300/portTICK_RATE_MS);
    stack_init();
    vTaskDelay(200/portTICK_RATE_MS);
    /* Temp init variable */
}
```

```

{
    stack_init_t stack_rules;
    start_status_t status;
    memset(stack_rules.mac_address, (uint8_t) 0, 8);

    memset(stack_rules.mac_address, (uint8_t) 0, 8);

    stack_rules.channel = CC2420_DEFAULT_CHANNEL;
    stack_rules.type = AD_HOC_DEVICE;
    for(i=0; i<8; i++)
    {
        stack_rules.mac_address[i] = mac_long.address[i];
    }

    vTaskDelay(300/portTICK_RATE_MS);
    status = stack_start(&stack_rules);
    if(status==START_SUCCESS)
    {
        //debug("Start Ok\r\n");
    }
}

for(j=0;j<45;j++)
{
    plr[j]=0;

    ssrp[j]=0;

    lqi[j]=0;
}

cipv6_compress_mode(1);
cudp_compress_mode(1);
debug("Task start.\r\n");
buffer = 0;
discover_socket = 0;
data=0;
P6DIR |= 0xC0;
P6DIR &= ~0x0F;
P6OUT |= 0xC0;
P5DIR &= ~0x80;
P2DIR &= ~0x01;
debug("Sensinode Micro_light_control_a500 Shell v1.0\r\n\r\n");

data_rx = socket(MODULE_CUDP, 0);    /* Socket for response data from port number 45 */

if (data_rx)
{
    if (socket_bind(data_rx, &data_address) != pdTRUE)
    {
        debug("Bind failed.\r\n");
    }
}
else

```

```
{
    debug("Data-RX socket creation failed.\r\n");
}
xLastWakeTime = xTaskGetTickCount();
LED1_ON();
vTaskDelayUntil( &xLastWakeTime, 1000 / portTICK_RATE_MS );
LED1_OFF();

rf_power_set(100);

for(;;)
{
    vTaskDelayUntil( &xLastWakeTime, 50 / portTICK_RATE_MS );
    if(count > 399) /* Send led control message request over network */
    {
        count=0;
        if (data_rx != 0)
        {
            buffer_t *buf = socket_buffer_get(data_rx);
            if (buf)
            {
                buf->options.hop_count = 5;
                buf->buf[buf->buf_end++] = SINK_REQUEST;
                buf->buf[buf->buf_end++] = LED_CONTROL;
                buf->buf[buf->buf_end++] = (discover_counter >> 8);
                buf->buf[buf->buf_end++] = (uint8_t) discover_counter;
                discover_address.port = 61619;
                if (socket_sendto(data_rx, &discover_address, buf) ==
pdTRUE)
                {
                    discover_counter++;
                }
                else
                {
                    debug("Discover send failed.\r\n");
                    socket_buffer_free(buf);
                    buf=0;
                }
            }
            else
            {
                debug("No buffer.\r\n");
            }
        }
        else
        {
            debug("No socket.\r\n");
        }
    }

    /* If pressed Buttom 1 */
    if (P2IN & 0x01)
    {
```

```

while(P2IN & 0x01){}

for (exper=0; exper<NUM Rondas; exper++)
{
    for (msg=0; msg<NUM_MSG; msg++)
    {
        vTaskDelay(1000/portTICK_RATE_MS);

        if (data_rx && sink_active == 1)
        {
            buffer_t *buf = socket_buffer_get(data_rx);
            if (buf)
            {
                buf->buf[buf->buf_end++] =
SENSOR_DATA;
                buf->buf[buf->buf_end++] =
BUTTON_1;
                if (socket_sendto(data_rx, &data_address,
buf) != pdTRUE)
                {
                    debug("Data send failed.\r\n");
                    socket_buffer_free(buf);
                    buf=0;
                }
            }
            else
            {
                debug("No buffer.\r\n");
            }
        }
        else
        {
            debug("No socket or SINK-device.\r\n");
        }
    }

    P6OUT ^= 0x40;
    vTaskDelay(5000/portTICK_RATE_MS);
    P6OUT ^= 0x40;
}

if (P5IN & 0x80)
{
    while(P5IN & 0x80){}

    count_power++;

    if (count_power == 4) count_power=0;
}

```



```
if (count_power == 0)
{
    tx_power = 100;

    rf_power_set(tx_power);
    P6OUT ^= 0x40;
    P6OUT ^= 0x80;
    vTaskDelay(300/portTICK_RATE_MS);
    P6OUT ^= 0x40;
    P6OUT ^= 0x80;

}

if (count_power == 1)
{
    tx_power = 60;

    rf_power_set(tx_power);

    P6OUT ^= 0x40;

    vTaskDelay(300/portTICK_RATE_MS);

    P6OUT ^= 0x40;

}

if (count_power == 2)
{
    tx_power = 30;

    rf_power_set(tx_power);

    P6OUT ^= 0x80;

    vTaskDelay(300/portTICK_RATE_MS);

    P6OUT ^= 0x80;

}

if (count_power == 3)
{
    tx_power = 10;

    rf_power_set(tx_power);

    P6OUT ^= 0x40;

    vTaskDelay(1000/portTICK_RATE_MS);

    P6OUT ^= 0x40;

}
```

```

    }

    /* Read Data socket */
    if (data_rx)
    {
        data = socket_read(data_rx, 10);
        if (data)
        {
            uint8_t ind=0, header=0;
            ind = data->buf_ptr;
            length = data->buf_end - data->buf_ptr;
            if (data->dst_sa.port == 61619)
            {
                header = data->buf[ind++];
                if( header == SINK_REQUEST)
                {
                    type = data->buf[ind++];
                    if(type == LED_CONTROL)
                    {
                        debug_printf("Rx SINK-REQUEST,
sensor data send activated, seq=%2.2x ",data->buf[ind + 1]);
                        debug("\r\n");
                        sink_active=1;
                        for(i=0; i < 10; i++)
                        {
                            data_address.address[i] =data-
>src_sa.address[i];
                        }
                    }
                }
            }
            socket_buffer_free(data);
            data = 0;
        }
    }
    count++;
}
}
}

```

## A.2 Nodo Receptor

```
/* Standard includes. */
#include <stdlib.h>
#include <signal.h>
#include <string.h>

/* Scheduler includes. */
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"

#include "bus.h"
#include "gpio.h"

#include "debug.h"

static void vapplitest( void *pvParameters );
#include "socket.h"
#include "rf.h"
#include "control_message.h"
#include "neighbor_routing_table.h"

/* Message types */
#define SINK_REQUEST 0x77
#define SENSOR_DATA 0xaa

/* Control types */
#define LED_CONTROL 0xe0
#define NORMAL_DATA 0xe1

/* Control attributes */
#define BUTTOM_1 0x10
#define BUTTOM_2 0x20

#define NUM_MSG 20
#define NUM_RONDAS 3

extern sockaddr_t mac_long;

sockaddr_t discover_address =
{
    ADDR_802_15_4_PAN_LONG,
    { 0xFF, 0xFF, 0xFF, 0xFF,
      0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF },
    61619
};

sockaddr_t data_address =
{
    ADDR_BROADCAST,
    { 0x00, 0x00, 0x00, 0x00,
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    61619
};
```

```

};

int main( void )
{
    LED_INIT();
    if (bus_init() == pdFALSE)
    {
    }

    xTaskCreate( vapplitest, "Term", 250, NULL, ( tskIDLE_PRIORITY + 1 ), NULL );
    vTaskStartScheduler();
    return 0;
}
/*-----*/
typedef uint8_t blw_reg[8];
buffer_t *buffer;
buffer_t *data;
socket_t *discover_socket=0, *control=0, *data_rx=0;
control_message_t *msg;
int8_t ping_active=0;
portTickType ping_start;

static void vapplitest( void *pvParameters )
{
    uint8_t i=0, sink_active=0, type=0, msg=0;
    uint8_t length, set_led;
    uint16_t discover_counter = 0, count=100;
    portTickType xLastWakeTime;
    pause(200);
    debug_init(115200);
    pause(300);

    uint8_t count_power=0, flag =0, sem=0, pos=0, exper=0;
    uint8_t tx_power=0;
    int16_t byte;

    int8_t j=0;

    int8_t plr[45];
    int16_t ssrc[45];
    int16_t lqi[45];

    vTaskDelay(300/portTICK_RATE_MS);
    stack_init();
    vTaskDelay(200/portTICK_RATE_MS);
    /* Temp init variable */
    {
        stack_init_t stack_rules;
        start_status_t status;
        memset(stack_rules.mac_address, (uint8_t) 0, 8);

        memset(stack_rules.mac_address, (uint8_t) 0, 8);
    }
}

```

```
stack_rules.channel = CC2420_DEFAULT_CHANNEL;
stack_rules.type = AD_HOC_DEVICE;
for(i=0; i<8; i++)
{
    stack_rules.mac_address[i] = mac_long.address[i];
}

vTaskDelay(300/portTICK_RATE_MS);
status = stack_start(&stack_rules);
if(status==START_SUCCESS)
{
    //debug("Start Ok\r\n");
}
}

for(j=0;j<45;j++)
{
    plr[j]=0;

    ssrp[j]=0;

    lqi[j]=0;
}

cipv6_compress_mode(1);
cudp_compress_mode(1);
debug("Task start.\r\n");
buffer = 0;
discover_socket = 0;
data=0;
P6DIR |= 0xC0;
P6DIR &= ~0x0F;
P6OUT |= 0xC0;
P5DIR &= ~0x80;
P2DIR &= ~0x01;
debug("Sensinode Micro_light_control_a500 Shell v1.0\r\n\r\n");

data_rx = socket(MODULE_CUDP, 0);    /* Socket for response data from port number 45 */

if (data_rx)
{
    if (socket_bind(data_rx, &data_address) != pdTRUE)
    {
        debug("Bind failed.\r\n");
    }
}
else
{
    debug("Data-RX socket creation failed.\r\n");
}

xLastWakeTime = xTaskGetTickCount();
LED1_ON();
```

```

vTaskDelayUntil( &xLastWakeTime, 1000 / portTICK_RATE_MS );
LED1_OFF();

rf_power_set(100);

for(;;)
{
    vTaskDelayUntil( &xLastWakeTime, 50 / portTICK_RATE_MS );
    if(count >399) /* Send led control message request over network */
    {
        count=0;
        if (data_rx != 0)
        {
            buffer_t *buf = socket_buffer_get(data_rx);
            if (buf)
            {
                buf->options.hop_count = 5;
                buf->buf[buf->buf_end++] = SINK_REQUEST;
                buf->buf[buf->buf_end++] = LED_CONTROL;
                buf->buf[buf->buf_end++] = (discover_counter >> 8);
                buf->buf[buf->buf_end++] = (uint8_t) discover_counter;
                discover_address.port = 61619;
                if (socket_sendto(data_rx, &discover_address, buf) ==
pdTRUE)
                {
                    discover_counter++;
                }
                else
                {
                    debug("Discover send failed.\r\n");
                    socket_buffer_free(buf);
                    buf=0;
                }
            }
            else
            {
                debug("No buffer.\r\n");
            }
        }
        else
        {
            debug("No socket.\r\n");
        }
    }

    /* If pressed Buttom 1*/
    if (P2IN & 0x01)
    {
        while(P2IN & 0x01){}

        pos++;
    }
}

```

```
byte = debug_read_blocking(2);

if(byte != -1)
{
    switch(byte)
    {
        case 'r':
            for(j=0;j<=pos;j++)
            {
                debug_int(plr[j]);
                debug("\r\n");
            }
            debug("\r\n");
            break;

        case 's':
            for(j=0;j<=pos;j++)
            {
                debug_int(ssrp[j]/plr[j]);
                debug("\r\n");
            }
            debug("\r\n");
            break;

        case 'q':
            for(j=0;j<=pos;j++)
            {
                debug_int(lqi[j]/plr[j]);
                debug("\r\n");
            }
            debug("\r\n");
            break;

        default:
            debug_put(byte);
            break;
    }
}

/* Read Data socket */
if (data_rx)
{
    data = socket_read(data_rx, 10);
    if (data)
    {
        uint8_t ind=0, header=0;
        ind = data->buf_ptr;
        length = data->buf_end - data->buf_ptr;
        if (data->dst_sa.port == 61619)
        {
            header = data->buf[ind++];
            if( header == SINK_REQUEST)
```

```

        {
            type = data->buf[ind++];
            if(type == LED_CONTROL)
            {
                debug_printf("Rx SINK-REQUEST,
sensor data send activated, seq=%2.2x ",data->buf[ind + 1]);
                debug("\r\n");
                sink_active=1;
                for(i=0; i < 10; i++)
                {
                    data_address.address[i] =data-
>src_sa.address[i];
                }
            }
            if( header == SENSOR_DATA)
            {
                set_led = data->buf[ind++];
                if(set_led==BUTTOM_1)
                {
                    plr[pos]=plr[pos]+1;
                    ssrcp[pos] = ssrcp[pos] + data-
>options.rf_dbm;

                    lqi[pos] = lqi[pos] + data->options.rf_lqi;
                    debug("expl ");

                    P6OUT ^= 0x40;
                    vTaskDelay(250/portTICK_RATE_MS);
                    P6OUT ^= 0x40;
                }
            }
        }
        socket_buffer_free(data);
        data = 0;
    }
    }
    count++;
}
}
}

```





# Anexo B

## Código del Experimento C

---

### B.1 Nodo Emisor

```
/* Standard includes. */
#include <stdlib.h>
#include <signal.h>
#include <string.h>

/* Scheduler includes. */
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"

#include "bus.h"
#include "gpio.h"

#include "debug.h"

static void vapplitest( void *pvParameters );
#include "socket.h"
#include "rf.h"
#include "control_message.h"
#include "neighbor_routing_table.h"

/* Message types */
#define SINK_REQUEST 0x77
#define SENSOR_DATA 0xaa

/* Control types */
#define LED_CONTROL 0xe0
#define NORMAL_DATA 0xe1
/* Control attributes */
#define BUTTOM_1 0x10
#define BUTTOM_2 0x20

#define NUM_MSG 10

extern sockaddr_t mac_long;

sockaddr_t discover_address =
{
    ADDR_802_15_4_PAN_LONG,
    { 0xFF, 0xFF, 0xFF, 0xFF,
      0xFF, 0xFF, 0xFF, 0xFF, 0xFF },
    61619
};

/* Address to Partner */
```

```
sockaddr_t data_address =
{
    ADDR_802_15_4_PAN_LONG,
    { 0x00, 0x00, 0x00, 0x00,
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    61619
};

int main( void )
{
    LED_INIT();
    if (bus_init() == pdFALSE)
    {
    }

    xTaskCreate( vapplitest, "Term", 250, NULL, ( tskIDLE_PRIORITY + 1 ), NULL );
    vTaskStartScheduler();
    return 0;
}
/*-----*/
typedef uint8_t b1w_reg[8];
buffer_t *buffer;
buffer_t *data;
socket_t *discover_socket=0, *control=0, *data_rx=0;
control_message_t *msg;
int8_t ping_active=0;
portTickType ping_start;
uint16_t total_sent=0, total_received=0;

//vTaskDelay(300/portTICK_RATE_MS);
static void vapplitest( void *pvParameters )
{
    uint8_t i=0, sink_active=0, type=0, msg=0;
    uint8_t length, set_led;
    uint16_t discover_counter = 0, count=100;
    portTickType xLastWakeTime;
    pause(200);
    debug_init(115200);
    pause(300);

    uint8_t count_power=0, flag =0;
    uint8_t count_mess=0, count_mess2=0;

    int16_t byte;
    uint8_t tx_power= CC2420_DEFAULT_POWER;

    vTaskDelay(300/portTICK_RATE_MS);
    stack_init();
    vTaskDelay(200/portTICK_RATE_MS);
    /* Temp init variable */
    {
        stack_init_t stack_rules;
```

```

start_status_t status;
memset(stack_rules.mac_address, (uint8_t) 0, 8);

memset(stack_rules.mac_address, (uint8_t) 0, 8);

stack_rules.channel = CC2420_DEFAULT_CHANNEL;
stack_rules.type = AD_HOC_DEVICE;
for(i=0; i<8; i++)
{
    stack_rules.mac_address[i] = mac_long.address[i];
}

vTaskDelay(300/portTICK_RATE_MS);
status = stack_start(&stack_rules);
if(status==START_SUCCESS)
{
    //debug("Start Ok\r\n");
}
}

cipv6_compress_mode(1);
cudp_compress_mode(1);
debug("Task start.\r\n");
buffer = 0;
discover_socket = 0;
data=0;
P6DIR |= 0xC0;
P6DIR &= ~0x0F;
P6OUT |= 0xC0;
P5DIR &= ~0x80;
P2DIR &= ~0x01;
debug("Sensinode Micro_light_control_a500 Shell v1.0\r\n\r\n");

data_rx = socket(MODULE_CUDP, 0);    /* Socket for response data from port number 45 */

if (data_rx)
{
    if (socket_bind(data_rx, &data_address) != pdTRUE)
    {
        debug("Bind failed.\r\n");
    }
}
else
{
    debug("Data-RX socket creation failed.\r\n");
}

xLastWakeTime = xTaskGetTickCount();
LED1_ON();
vTaskDelayUntil( &xLastWakeTime, 1000 / portTICK_RATE_MS );
LED1_OFF();

for(;;)

```

```
{
    vTaskDelayUntil( &xLastWakeTime, 50 / portTICK_RATE_MS );
    if(count >399) /* Send led control message request over network */
    {
        count=0;
        if (data_rx != 0)
        {
            buffer_t *buf = socket_buffer_get(data_rx);
            if (buf)
            {
                buf->options.hop_count = 5;
                buf->buf[buf->buf_end++] = SINK_REQUEST;
                buf->buf[buf->buf_end++] = LED_CONTROL;
                buf->buf[buf->buf_end++] = (discover_counter >> 8);
                buf->buf[buf->buf_end++] = (uint8_t) discover_counter;
                discover_address.port = 61619;
                if (socket_sendto(data_rx, &discover_address, buf) ==
pdTRUE)
                {
                    discover_counter++;
                }
                else
                {
                    debug("Discover send failed.\r\n");
                    socket_buffer_free(buf);
                    buf=0;
                }
            }
            else
            {
                debug("No buffer.\r\n");
            }
        }
        else
        {
            debug("No socket.\r\n");
        }
    }

    /* If pressed Button 1 */
    if (P2IN & 0x01)
    {
        while(P2IN & 0x01){}

        if (flag == 0) flag = 1;
        else flag = 0;
    }

    if(flag)
    {
        for (msg=0; msg<NUM_MSG; msg++)
        {
```

```

vTaskDelay(93.5/portTICK_RATE_MS);

if (data_rx && sink_active == 1)
{
    buffer_t *buf = socket_buffer_get(data_rx);
    if (buf)
    {
        buf->buf[buf->buf_end++] = SENSOR_DATA;
        buf->buf[buf->buf_end++] = BUTTOM_1;

        if (socket_sendto(data_rx, &data_address, buf) !=
pdTRUE)
        {
            debug("Data send failed.\r\n");
            socket_buffer_free(buf);
            buf=0;
        }
        else
        {
            //count_mess++;
            total_sent++;
        }
    }
    else
    {
        debug("No buffer.\r\n");
    }
}
else
{
    debug("No socket or SINK-device.\r\n");
}
}

byte = debug_read_blocking(2);

if(byte != -1)
{
    switch(byte)
    {
        case 's':
            debug_int(total_sent);
            debug(" messages sent.\r\n");
            break;

        case '+':
            if(tx_power==100)
            {
                debug("Max Tx power set up.\r\n");
            }
        }
    }
}

```

```
        }
        else
        {
            tx_power += 10;
            rf_power_set(tx_power);
            debug_printf("Increase Tx power, current state
%d.\r\n", tx_power);
        }
        break;

    case '-':
        if(tx_power==10)
        {
            debug("Min Tx power set up 10.\r\n");
        }
        else
        {
            tx_power -= 10;
            rf_power_set(tx_power);
            debug_printf("Decrease Tx power, current state
%d.\r\n", tx_power);
        }
        break;

    case 'r':
        debug("\r\n");
        break;

    default:
        debug_put(byte);
        break;
    }
}

/* Read Data socket */
if (data_rx)
{
    data = socket_read(data_rx, 10);
    if (data)
    {
        uint8_t ind=0, header=0;
        ind = data->buf_ptr;
        length = data->buf_end - data->buf_ptr;
        if (data->dst_sa.port == 61619)
        {
            header = data->buf[ind++];
            if (header == SINK_REQUEST)
            {
                type = data->buf[ind++];
                if(type == LED_CONTROL)
                {
                    debug_printf("Rx SINK-REQUEST,
sensor data send activated, seq=%2.2x ",data->buf[ind + 1]);
                    debug("\r\n");
                }
            }
        }
    }
}
```





## B.2 Nodo Receptor

```
/* Standard includes. */
#include <stdlib.h>
#include <signal.h>
#include <string.h>

/* Scheduler includes. */
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"

#include "bus.h"
#include "gpio.h"

#include "debug.h"

static void vapplitest( void *pvParameters );
#include "socket.h"
#include "rf.h"
#include "control_message.h"
#include "neighbor_routing_table.h"

/* Message types */
#define SINK_REQUEST 0x77
#define SENSOR_DATA 0xaa

/* Control types */
#define LED_CONTROL 0xe0
#define NORMAL_DATA 0xe1

/* Control attributes */
#define BUTTOM_1 0x10
#define BUTTOM_2 0x20

#define NUM_MSG 10

extern sockaddr_t mac_long;

sockaddr_t discover_address =
{
    ADDR_802_15_4_PAN_LONG,
    { 0xFF, 0xFF, 0xFF, 0xFF,
      0xFF, 0xFF, 0xFF, 0xFF, 0xFF },
    61619
};

/* Address to Partner */
sockaddr_t data_address =
{
    ADDR_802_15_4_PAN_LONG,
    { 0x00, 0x00, 0x00, 0x00,
      0x00, 0x00, 0x00, 0x00, 0x00 },
}
```

```

        61619
    };

int main( void )
{
    LED_INIT();
    if (bus_init() == pdFALSE)
    {
    }

    xTaskCreate( vapplitest, "Term", 250, NULL, ( tskIDLE_PRIORITY + 1 ), NULL );
    vTaskStartScheduler();
    return 0;
}
/*-----*/
typedef uint8_t blw_reg[8];
buffer_t *buffer;
buffer_t *data;
socket_t *discover_socket=0, *control=0, *data_rx=0;
control_message_t *msg;
int8_t ping_active=0;
portTickType ping_start;
uint16_t total_sent=0, total_received=0;

static void vapplitest( void *pvParameters )
{
    uint8_t i=0, sink_active=0, type=0, msg=0;
    uint8_t length, set_led;
    uint16_t discover_counter = 0, count=100;
    portTickType xLastWakeTime;
    pause(200);
    debug_init(115200);
    pause(300);

    uint8_t count_power=0, flag =0;
    uint8_t count_mess=0, count_mess2=0;

    int16_t byte;
    uint8_t tx_power= CC2420_DEFAULT_POWER;

    vTaskDelay(300/portTICK_RATE_MS);
    stack_init();
    vTaskDelay(200/portTICK_RATE_MS);
    /* Temp init variable */
    {
        stack_init_t stack_rules;
        start_status_t status;
        memset(stack_rules.mac_address, (uint8_t) 0, 8);

        memset(stack_rules.mac_address, (uint8_t) 0, 8);
    }
}

```

```
stack_rules.channel = CC2420_DEFAULT_CHANNEL;
stack_rules.type = AD_HOC_DEVICE;
for(i=0; i<8; i++)
{
    stack_rules.mac_address[i] = mac_long.address[i];
}

vTaskDelay(300/portTICK_RATE_MS);
status = stack_start(&stack_rules);
if(status==START_SUCCESS)
{
    //debug("Start Ok\r\n");
}
}

cipv6_compress_mode(1);
cudp_compress_mode(1);
debug("Task start.\r\n");
buffer = 0;
discover_socket = 0;
data=0;
P6DIR |= 0xC0;
P6DIR &= ~0x0F;
P6OUT |= 0xC0;
P5DIR &= ~0x80;
P2DIR &= ~0x01;
debug("Sensinode Micro_light_control_a500 Shell v1.0\r\n\r\n");

data_rx = socket(MODULE_CUDP, 0); /* Socket for response data from port
number 45 */

if (data_rx)
{
    if (socket_bind(data_rx, &data_address) != pdTRUE)
    {
        debug("Bind failed.\r\n");
    }
}
else
{
    debug("Data-RX socket creation failed.\r\n");
}

xLastWakeTime = xTaskGetTickCount();
LED1_ON();
vTaskDelayUntil( &xLastWakeTime, 1000 / portTICK_RATE_MS );
LED1_OFF();

for(;;)
{
    vTaskDelayUntil( &xLastWakeTime, 50 / portTICK_RATE_MS );
    if(count >399) /* Send led control message request over network */
    {
```

```

count=0;
if (data_rx != 0)
{
    buffer_t *buf = socket_buffer_get(data_rx);
    if (buf)
    {
        buf->options.hop_count = 5;
        buf->buf[buf->buf_end++] = SINK_REQUEST;
        buf->buf[buf->buf_end++] = LED_CONTROL;
        buf->buf[buf->buf_end++] = (discover_counter >> 8);
        buf->buf[buf->buf_end++] = (uint8_t) discover_counter;
        discover_address.port = 61619;
        if (socket_sendto(data_rx, &discover_address, buf) ==
pdTRUE)
        {
            discover_counter++;
        }
        else
        {
            debug("Discover send failed.\r\n");
            socket_buffer_free(buf);
            buf=0;
        }
    }
    else
    {
        debug("No buffer.\r\n");
    }
}
else
{
    debug("No socket.\r\n");
}
}

byte = debug_read_blocking(2);

if(byte != -1)
{
    switch(byte)
    {
        case 'r':
            debug_int(total_received);
            debug(" messages received.\r\n");
            break;

        case '\r':
            debug("\r\n");
            break;

        default:
            debug_put(byte);
            break;
    }
}

```

```
    }
}
/* Read Data socket */
if (data_rx)
{
    data = socket_read(data_rx, 10);
    if (data)
    {
        uint8_t ind=0, header=0;
        ind = data->buf_ptr;
        length = data->buf_end - data->buf_ptr;
        if (data->dst_sa.port == 61619)
        {
            header = data->buf[ind++];
            if( header == SINK_REQUEST)
            {
                type = data->buf[ind++];
                if(type == LED_CONTROL)
                {
                    debug_printf("Rx SINK-REQUEST,
sensor data send activated, seq=%2.2x ",data->buf[ind + 1]);
                    debug("\r\n");
                    sink_active=1;
                    for(i=0; i < 10; i++)
                    {
                        data_address.address[i] =data-
>src_sa.address[i];
                    }
                }
            }
            if( header == SENSOR_DATA)
            {
                set_led = data->buf[ind++];
                if(set_led==BUTTOM_1)
                {
                    total_received++;
                }
            }
        }
        socket_buffer_free(data);
        data = 0;
    }
}
count++;
}
}
```

## B.3 Nodo Incomunicado

```

/* Standard includes. */
#include <stdlib.h>
#include <signal.h>
#include <string.h>

/* Scheduler includes. */
#include "FreeRTOS.h"
#include "task.h"
#include "queue.h"
#include "semphr.h"

#include "bus.h"
#include "gpio.h"

#include "debug.h"

static void vapplitest( void *pvParameters );
#include "socket.h"
#include "rf.h"
#include "control_message.h"
#include "neighbor_routing_table.h"

/* Message types */
#define SINK_REQUEST 0x77
#define SENSOR_DATA 0xaa

/* Control types */
#define LED_CONTROL 0xe0
#define NORMAL_DATA 0xe1

/* Control attributes */
#define BUTTOM_1 0x10
#define BUTTOM_2 0x20

#define NUM_MSG 10

extern sockaddr_t mac_long;

sockaddr_t discover_address =
{
    ADDR_802_15_4_PAN_LONG,
    { 0xFF, 0xFF, 0xFF, 0xFF,
      0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF },
    61619
};

/* Address to Partner */
sockaddr_t data_address =
{
    ADDR_802_15_4_PAN_LONG,
    { 0x00, 0x00, 0x00, 0x00,

```

```
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    61619
};

int main( void )
{
    LED_INIT();
    if (bus_init() == pdFALSE)
    {
    }

    vTaskCreate( vapplitest, "Term", 250, NULL, ( tskIDLE_PRIORITY + 1 ), NULL );
    vTaskStartScheduler();
    return 0;
}
/*-----*/
typedef uint8_t b1w_reg[8];
buffer_t *buffer;
buffer_t *data;
socket_t *discover_socket=0, *control=0, *data_rx=0;
control_message_t *msg;
int8_t ping_active=0;
portTickType ping_start;
uint16_t total_sent=0, total_received=0;

//vTaskDelay(300/portTICK_RATE_MS);
static void vapplitest( void *pvParameters )
{
    uint8_t i=0, sink_active=0, type=0, msg=0;
    uint8_t length, set_led;
    uint16_t discover_counter = 0, count=100;
    portTickType xLastWakeTime;
    pause(200);
    debug_init(115200);
    pause(300);

    uint8_t count_power=0, flag =0;
    uint8_t count_mess=0, count_mess2=0;

    int16_t byte;
    uint8_t tx_power= CC2420_DEFAULT_POWER;

    vTaskDelay(300/portTICK_RATE_MS);
    stack_init();
    vTaskDelay(200/portTICK_RATE_MS);
    /* Temp init variable */
    {
        stack_init_t stack_rules;
        start_status_t status;
        memset(stack_rules.mac_address, (uint8_t) 0, 8);

        memset(stack_rules.mac_address, (uint8_t) 0, 8);
    }
}
```

```

        stack_rules.channel = CC2420_DEFAULT_CHANNEL;
        stack_rules.type = AD_HOC_DEVICE;
        for(i=0; i<8; i++)
        {
            stack_rules.mac_address[i] = mac_long.address[i];
        }

        vTaskDelay(300/portTICK_RATE_MS);
        status = stack_start(&stack_rules);
        if(status==START_SUCCESS)
        {
            //debug("Start Ok\r\n");
        }
    }

    cipv6_compress_mode(1);
    cudp_compress_mode(1);
    debug("Task start.\r\n");
    buffer = 0;
    discover_socket = 0;
    data=0;
    P6DIR |= 0xC0;
    P6DIR &= ~0x0F;
    P6OUT |= 0xC0;
    P5DIR &= ~0x80;
    P2DIR &= ~0x01;
    debug("Sensinode Micro_light_control_a500 Shell v1.0\r\n\r\n");

    data_rx = socket(MODULE_CUDP, 0);    /* Socket for response data from port number 45 */

    if (data_rx)
    {
        if (socket_bind(data_rx, &data_address) != pdTRUE)
        {
            debug("Bind failed.\r\n");
        }
    }
    else
    {
        debug("Data-RX socket creation failed.\r\n");
    }

    xLastWakeTime = xTaskGetTickCount();
    LED1_ON();
    vTaskDelayUntil( &xLastWakeTime, 1000 / portTICK_RATE_MS );
    LED1_OFF();

    for(;;)
    {
        vTaskDelayUntil( &xLastWakeTime, 50 / portTICK_RATE_MS );
        if(count >399) /* Send led control message request over network */
        {

```



```
count=0;
if (data_rx != 0)
{
    buffer_t *buf = socket_buffer_get(data_rx);
    if (buf)
    {
        buf->options.hop_count = 5;
        buf->buf[buf->buf_end++] = SINK_REQUEST;
        buf->buf[buf->buf_end++] = LED_CONTROL;
        buf->buf[buf->buf_end++] = (discover_counter >> 8);
        buf->buf[buf->buf_end++] = (uint8_t) discover_counter;
        discover_address.port = 61619;
        //if (socket_sendto(data_rx, &discover_address, buf) ==
pdTRUE)
        //{
        //    discover_counter++;
        //}
        //else
        //{
            debug("Discover send failed.\r\n");
            socket_buffer_free(buf);
            //    buf=0;
            //}
        }
    else
    {
        debug("No buffer.\r\n");
    }
}
else
{
    debug("No socket.\r\n");
}
}

/* If pressed Button 1*/
if (P2IN & 0x01)
{
    while(P2IN & 0x01){}

    //vTaskDelay(8000/portTICK_RATE_MS);
    if (flag == 0) flag = 1;
    else flag = 0;
}

if(flag)
{
    debug("It is sending.\r\n");
    for (msg=0; msg<NUM_MSG; msg++)
    {

        vTaskDelay(93.5/portTICK_RATE_MS);
    }
}
```

```

        if (data_rx && sink_active == 1)
        {
            buffer_t *buf = socket_buffer_get(data_rx);
            if (buf)
            {
                buf->buf[buf->buf_end++] = SENSOR_DATA;
                buf->buf[buf->buf_end++] = BUTTOM_1;

                //buf->size=total_sent

                //if (socket_sendto(data_rx, &data_address, buf) !=
pdTRUE)
                //{
                //    debug("Data send failed.\r\n");
                //    socket_buffer_free(buf);
                //    buf=0;
                //}
                //else
                //{
                //    //count_mess++;
                //    total_sent++;
                //}
            }
            else
            {
                debug("No buffer.\r\n");
            }
        }
        else
        {
            debug("No socket or SINK-device.\r\n");
        }
    }

    /* Read Data socket */
    if (data_rx)
    {
        data = socket_read(data_rx, 10);
        if (data)
        {
            debug("It is receiving.\r\n");
            uint8_t ind=0, header=0;
            ind = data->buf_ptr;
            length = data->buf_end - data->buf_ptr;
            if (data->dst_sa.port == 61619)
            {
                header = data->buf[ind++];
                if (header == SINK_REQUEST)
                {
                    type = data->buf[ind++];
                }
            }
        }
    }
}

```



# Referencias

---

- [1] Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, Tian He, y John A. Sankovic: "ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks", *SenSys'06*, Noviembre 1-3, 2006, Boulder, Colorado, USA.
- [2] Jerry Zhao y Ramesh Govindan: "Understanding Packet Delivery Performance In Dense Wireless Sensor Networks", conferencia *SenSys'03*, Noviembre 5-7, 2003, Los Angeles, California, USA.
- [3] M. Horton, D. Culler, K. Pister, J. Hill, R. Szewczyk, y A. Woo: "MICA, The Commercialization of Microsensor Motes", en *Sensors Magazine*, páginas 40–48, Abril del 2002.
- [4] S. Blom, C. Bellettini, A. Sinigalliesi, L. Stabellini, M. Rossi, y G. Mazzini: "Transmission Power Measurements for Wireless Sensor Nodes and their Relationship to the Battery Level", 2005.
- [5] Proyecto Eyes: "EYES: Energy Efficient Wireless Sensor Networks", [www.eyes.eu.org](http://www.eyes.eu.org)
- [6] Edgar H. Callaway Jr.: "Wireless sensor networks: architectures and protocols", páginas 1-11, 2004.
- [7] Khanh Tuan Le: "Designing a ZigBee-ready IEEE 802.15.4-compliant radio transceiver", Noviembre, 2004.
- [8] Sensinode Ltd., <http://www.sensinode.com>
- [9] Sensinode Ltd.: "OEM product catalog", 2007.
- [10] Sensinode Ltd.: "NanoStack Manual v1.0.0", 2007.
- [11] IEEE 802.15.4 Standard, <http://www.ieee.org>
- [12] 6LoWPAN Standard, RFC 4944, <http://www.ietf.org>
- [13] Sensinode Ltd.: "Micro Hardware Manual", 2006.
- [14] Crossbow Technology: "TelosB datasheet".
- [15] Crossbow Technology: "MIB510 datasheet".
- [16] Crossbow Technology: "MicaZ datasheet".

- [17] Crossbow Technology: "Mica2 datasheet".
- [18] Crossbow Technology: "Mica2Dot datasheet".
- [19] "Transmission Control Protocol", RFC 793, Septiembre de 1981
- [20] J. Postel: "User Datagram Protocol", RFC 768, 28 de Agosto de 1980
- [21] C. Wang, K. Sohraby, Bo Li y W. Tang: "Issues of Transport Control Protocols for Wireless Sensor Networks", Universidad de Arkansas, Fayetteville, AR, EEUU.
- [22] P. Pereira, A. Grilo, F. Rocha, M.S. Nunes, A. Casaca, C. Chaudet, P. Almstrom y M. Johansson: "End-to-End Reliability in WSNs: Survey and Research Challenges", Inescid Lisboa, Diciembre de 2007.
- [23] A. Cerpa, J.L. Wong, L. Kuang, M. Potkonjak y D. Estrin: "Statistical Model of Lossy Links in Wireless Sensor Networks", Universidad de California, Los Angeles, EEUU.
- [24] M. Mallinson, P. Drane y S. Hussain: "Discrete Radio Power Level Consumption Model in Wireless Sensor Networks", Universidad de Acadia, Nova Scotia, Canada, 2007.
- [25] C. Park, K. Lahiri y A. Raghunathan: "Battery Discharge Characteristics of Wireless Sensor Nodes: An Experimental Analysis", 2005.
- [26] Fractus S.A.: "Fractus Slim Reach Xtend Chip Antenna User Manual", Noviembre 2006.

# A technique for instrumentation of Wireless Sensor Network nodes

Javier Alfonso Aviles Garcia; A762185

ELD030

**Abstract:** In recent years, the great technological advance together with the desire to monitor all kind of systems and scenarios, has caused the appearance and fast development of the Wireless Sensor Networks (WSNs). Currently, the designer of such a network can acquire numerous different sets of wireless sensors, whose datasheets sometimes do not contain enough specifications to know their behaviour in a actual WSN. Hence, it is necessary to create a procedure which allow designers to instrument the devices's capabilities and can discover how they will behave when the system is deployed. This paper describes a systematic technique, which is divided into three experiments, in order to help the WSN creator to achieve that aim. Parameters such as the reception rate, the received signal strength, Link Quality Indicator (LQI), the orientation angle and the battery life are tested in depth by this technique. Furthermore, it is executed about a specific device and the results are shown in this paper as well.

## 1. Introduction

The collection of information from an environment to be able to control and oversee any activity is crucial. The data are obtained from very many different locations, some of them are very hostile. As a result, a wired set of sensors would not be appropriate in these areas. Hence Wireless Sensor Networks (WSNs) were created some few years ago. These are constituted by a significant number of tiny, low-cost and battery-powered devices, referred to as nodes, which sense several physical parameters, such as temperature and light. Nodes act autonomously but are part of a large distributed control system that has a specific common function as its target [1], for instance tracking inventory items in warehouses.

The fields where a Wireless Sensor Network could be utilized are very numerous. Some of them are: Intelligent Agriculture and Environmental Sensing, Industrial Control and Monitoring, Home Automation and Consumer Electronics, Security and Military Sensing, Asset Tracking and Supply Chain Management or Health Monitoring [2].

In spite of the great advances in the telecommunications and electrical industries in the last years, wireless communication is still very unpredictable, which causes major uncertainty when a WSN is used. In addition, the ad-hoc feature which the majority of the WSNs have, as well as the lack of standards and simulation tools, intensifies and makes the development of this kind of networks more complex.

The introduction of this paper continues with two subsections where the motivation behind this study is found as well as a review of other work in the area. In Section 2, the methodology to complete the instrumentation of the WSN nodes is described. The results obtained are in Section 3. Finally, the main conclusions are presented in Section 4.

### 1.1 Motivation

Currently many different sets of wireless sensors from a multitude of firms are available to be acquired by users. A significant quantity of these collections of devices are not able to communicate with each other as they do not have similar characteristics and also because they operate with different protocols. Normally, the nodes are deployed in very different scenarios to constitute particular applications (ad-hoc). Nevertheless, the sensors' datasheet do not specify the majority of their characteristics, and so their behaviour when they are working in a real environment is unpredictable. As a result of this fact, the existence of a methodology that lets the network creator know the behaviour of the devices in the real environment where they will be located is a necessary condition. For that, the devices must be deployed and tested physically in the place where the application will run, which involves a large effort. As in plenty of telecommunication systems, the simulation process would simplify this work. However the simulated results obtained in WSNs show a dependence on various theoretical assumptions which are not generally accurate enough to represent the correct network behaviour in a real environment, since it depends on some random and unmanageable factors.

Zhao and Govindan [3] defined methodology as a set of organized techniques to research the performance of WSNs in several scenarios. It supplies the users with information, which after being analysed, lets them specify the characteristics and behaviour of the device. This process is called *instrumentation* of a wireless sensor. When the group of techniques to

experiment with the nodes is described, the features which need to be tested must be listed. Some examples could be: packet loss rate, battery life, maximum reach, received signal strength performance, optimum orientation angle and error rate. These allow the network's designer to know the limitations of his devices which together with the specifications and requirements (quality criteria) of the system, will let the creator configure a network to carry out the intended functions. By way of illustration, two different examples:

- On the one hand, a user would like to deploy a WSN in a specific area to sense climate's variables. He has a high budget and firstly he decides to purchase some nodes to execute their instrumentation. After it, he finds out that the communication among their wireless sensors in this environment is quite poor because factors like the maximum reach (with the highest transmission power) and the battery life are too low. Therefore, he should buy numerous devices to occupy all the zone because he will need to locate them very near each other since the maximum reach is short. In addition, due to the poor battery life feature of the sensor, he should configure them to work with a low transmission power and so the network is able to have a longer life.
- On the other hand, another user with a low budget desires to create a network in a different area. After making the sensor instrumentation he discovers that the battery life and the maximum reach are not a problem in the deployment of his network and so he does not need to purchase so many devices to cover the region. They may be located far from each other and also may be configured to operate with high transmission power.

Achieving instrumentation of a set of sensors which will constitute a WSN, before they are deployed, is vital. Even if it was made to provide another network previously, it must be executed again as the specifications, requirements and variables of the new scenario may have changed. Variables means these factors whose alteration induces a change in the behaviour of a device. They can be classified in two main groups: manageable and unmanageable (random) factors. The first one is formed by these variables which the WSN's creator can control (transmission power, distance between devices, location, traffic packet distribution, ...), and the factors of the second one are all those which he cannot, such as weather or manufacturing flaws of the sensors. The methodology must determine which factors will be studied and altered in the tests to find out every characteristic of the wireless sensors' behaviour.

## 1.2 Related work

Numerous papers about wireless sensor instrumentation have been written in the past few years. However, just some of them carry out a wide test about the wireless sensors' behaviour. Furthermore, papers where a broad and detailed description of the instrumentation techniques is reported, are very scarce. In 2006, several scientists, such as Lin and Zhang [4], achieved systematic experiments with a set of wireless sensor and concluded that the radio communication quality in this kind of devices has a great instability with environment and time. In spite of the fact that they test deeply some parameters of the radio link, like the received signal strength or LQI (Link Quality Indicator), which are researched in this paper as well, their investigation is quite different because the technique to test the devices is not described in depth.

More related to our work, in 2003 Zhao and Govindan [3] launched a study about Mica nodes [5] where the packet reception performance of this kind of devices was researched. Basically, findings from packet loss rate experiments at both Physical and Medium Access Layer are explained. In addition, every test is carried out over differing environments, just as it is achieved in this paper. In summary, they declare that "gray areas" exist in the communication among the Mica nodes and that their behaviour in every scenario is quite changeable, though they do not give any explanation of these findings.

Many others works exist where a single feature of a wireless sensor (often the battery life) is tested and analysed thoroughly. For instance, [6] describes the ratio between the battery level and the emitted power in the EYESIFXv2 Infineon wireless sensor produced within the European EYES project [7]. Just as in this paper, the experimental setup is systematically detailed. It occurs in an indoor environment where five receivers, which are wired to different computers to store data, receive messages from a sender. Finally, it is noted that the relationship between transmission power and battery level is non linear, and hence non trivial.

## 2 Methodology

In this section one methodology to test the behaviour of any wireless sensor is described. It is composed by simple techniques which will allow WSN's designer to find out how his devices will work in a real performance. That behaviour is defined by the different characteristics of the sensor which can be very numerous. Because of that, this methodology just covers a range of them which are analysed in depth in the results section. These features are: packet loss rate, reception

rate, received signal strength, error rate, optimum orientation and battery life. Collision rate or ambient temperature behaviour would be some examples of non tested characteristics.

On the other hand, any communication system is affected by numerous phenomena (such as transmission power, environment, distance among devices, traffic packet distribution, communication protocol, weather, etc.), which are divided in two main groups: manageable and unmanageable variables, as explained previously. The variation of every factor causes a change in the behaviour of the wireless sensor, but the excessive number of different phenomena hinders us from testing all them. In addition, it is very complex to isolate the unmanageable ones to be able to investigate their impact on the devices behaviour, and so just a few manageable variables have been selected to carry out the experiments:

- **Distance:** the maximum reach which a wireless sensor can send packets to another one, is one of the most important parameters that the designer must know when a WSN is going to be deployed. To discover it, the distance between sensors is varied in the experiments. In principle the greater the reach, the lower the number of nodes utilized to occupy the whole region, which is an excellent advantage for those applications where few devices can be purchased due to their low budget. The decrease of the traffic in the network would be another advantage.
- **Transmission power:** what power the devices will use to send the messages, is a crucial decision for the WSN creator. Supposedly, the higher this power, the greater the maximum reach, but the sensor battery life could be affected, and it would be possibly shorter if the transmission power is increased. Consequently, this factor has an essential role in both battery life and maximum reach experiments.
- **Environment:** normally where the WSN is going to run, is not chosen directly by the designer because it is a factor which is bound up with the network application. According to [3] and [4], the devices show a quite different behaviour depending on the area where they are situated. Furthermore, the sort of environment defines the phenomena that influence in the wireless communication, and hence in the devices' characteristics as well. The three environments selected in order to complete the tests are:

*Corridor:* a high number of WSN applications (for instance the system to localize elderly in nursing homes) work in an indoor placement. Therefore, defining how the instrumentation of a specific sensor must be executed correctly in this kind of scenario has a significant importance. In addition, it is likely that the multi-path reflections due to the walls affect the results obtained. Our corridor belongs to a building of offices and its dimensions are 2m by 45m.

*Parking lot:* it is a 50m by 10m outside area where applications, such as a system to control some traffic lights and the speed of the vehicles, could run.

*Grass field:* the third experimental round is executed in another outside environment, whose size is 50m by 50m. In order to monitor both animals and plants habitats, a large sum of WSNs are deployed on this kind of placement. As on the parking lot area, the tests implemented in this region are influenced by the weather, whose random nature adds more uncertainty to the experiments.

In this work, the main aim was to complete the different experiments on an indoor as well as on an outdoor environment at least. The grass field area was the first outside area selected because of the numerous applications that run in this kind of surface currently. Furthermore, to study widely the possible effects of the multi-path reflections, a narrow corridor was chosen as the inside placement. Finally, a second outside area as the parking lot let us investigate the wireless sensor behaviour in another surface in order to study the possible differences.

- **Traffic packet distribution:** the data flow which a node receives and sends within its WSN must be characterized. In many applications the wireless sensors capture and transmit relevant information periodically to their neighbours, which repeat the same process. Finally, all the data are collected by a base-station (BS) that analyses them and later executes the appropriate decisions. Thus, the quantity of packets flowing through the network for every temporal window is fixed. That traffic packet distribution, which is utilized in our experiments, is called *uniform*. However, on other systems the devices, instead of sending periodical messages, sense and transmit information after certain events in the environment occur, consequently the network overload may be different at each moment in these scenarios.
- **Communication protocol:** as in most WSN and in the tests shown in this paper, the UDP transport layer protocol [8] is the technique which allows the wireless communication to be successful, though it does not guarantee delivery or



ordering of the messages. Various reasons exist so that this sort of networks works with UDP rather than TCP [9]. For instance, using a protocol like TCP which re-transmits the discarded packets due to having implemented an acknowledgement service, is a great disadvantage from the point of view of the battery life, since it increases the number of processed packets in the nodes and so the waste of energy is higher.

- **Number of wireless sensors:** when a designer decides to create a WSN, before purchasing all the devices, he should finish the instrumentation of the sensor in order to check if those devices fit the requirements, characteristics and quality criteria of the system. Consequently, at the beginning, just a few of them ought to be acquired, avoiding in this way, a decrease of the project budget whether those devices are not eventually selected. As a result of this fact, only 2 devices (sender and receiver) have been used in all experiments, except for the optimum orientation experiment where 4 sensors exchange messages. This decision reduces the experimental rigor of our sensor instrumentation because normally the WSN's nodes receive/generate traffic from/towards other ones, but, since the collision study is very complex and the set of devices available is quite scant, just a pair of wireless sensors has been utilized.

The methodology to instrument a wireless sensor has been divided in three main experiments according to the similarity of their execution process and their tested features. By means of the Experiment A, characteristics of the device such as the maximum reach for every transmission power, the received signal strength performance or how LQI varies with both distance and power, are tested. Next, to find out the behaviour of the node when its angle with regard to the another device changes, the Experiment B must be completed. Finally, the sensor's battery study is accomplished in the Experiment C. The value of every factor in the three experiments is summarized at Table 1.

Experiment	Distance	Trans. Power	Environment	Distribution	Protocol	N° Nodes
A	Variable	Variable	3	Uniform	UDP	2
B	Variable	Fixed	3	Uniform	UDP	2 / 4
C	Fixed	Variable	Corridor	Uniform	UDP	2

Table 1. Values of the different factors for every experiment.

## 2.1 Experiment A

How packet delivery performance, received signal strength and LQI varies with the distance, transmission power and environment is examined in this experiment. In this section, at the beginning, a brief description of the tested wireless sensor characteristics is presented. Later, our methodology is defined in detail in order to perform the experiment in a practical way.

### 2.1.1 Tested parameters

After the Experiment A is executed, the next three device's characteristics will be known completely by the WSN designer:

- **Reception rate:** one of the most important parameters at the behaviour of a wireless sensor is its received packet performance. Any communication system is constituted by components that generate traffic which may be delivered successfully or by contrast may be missed. In addition, if the protocol communication, such as UDP, does not implement any packet re-transmission service, the quantity of non received information by the nodes might be significant. As a result of this fact, initially when a WSN is planned, the received packet performance of the nodes must be examined in order to configure and deploy the network attempting to decrease the number of discarded and missed packets. Consequently, the *reception rate* is defined as the fraction of messages sent and successful delivered in a temporal window. Otherwise, the fraction of traffic transmitted but not delivered eventually constitutes the *packet loss rate*. These parameters are tested in this experiment while both distance and transmission power values are modified.
- **Received signal strength:** this is calculated, for each received message, measuring the signal power which is averaged over eight symbol periods. As a result of this, a very useful parameter is obtained. It may help to define the quality of the communication link as well as other numerous facts, such as finding out the node's sensibility so that it can receive a packet. It means, ascertaining what is the lowest signal power that the device must sense to be able to

receive information. The received signal strength's measure unit utilized in these experiments is the dBm.

- **Link Quality Indicator (LQI):** this is a metric of the current quality of the incoming signal. For each received message, it measures the *chip error rate* testing the error rate for the first eight symbols. In fact, LQI [10] is a relative measurement of the link quality (a high value means a low error rate and so a better link), and it is dependent on the modulation format.

### 2.1.2 Detailed experiment

With the aim of achieving a correct Experiment A execution and so getting the functions of the three previous characteristic quoted, the next procedure was strictly completed:

1. Load both transmission and reception application in the devices' memory. On one hand, the transmitter's software must create 20 UDP packets and must send them with a rate of 1 packet per second (uniform distribution). The device has to be configured so that these packets can be sent by the user indefinite times, for instance a switch which activates the sending. Furthermore, the transmission power value should be monitored by software as well. On the other hand, the receiver's software must be typed to capture and count the packets which come from the source. Moreover, it must store both received signal strength and LQI value of every incoming packet in data structures (for example arrays). Configuring both devices to light different sensor leds while the tests are running is advised because it may help to the user at the experimental process.
2. Fix with adhesive tape, to avoid movements that distort the experiment, both wireless sensors to different boxes. whose size has to be wide and long enough so that the sensor can be stably situated on it, and high enough so that no object (for instance the grass) can disturb the free link between transmitter and receiver. The boxes selected for our experiment have a size of 330 mm x 200 mm x 75 mm.
3. The measurements are collected in the three environments: corridor, parking lot and grass field; having been utilized unlike tools to create the metric scale in every scenario:

*Corridor:* small marks (for example with adhesive tape) each 2 metres are made on the floor or on the wall through all the corridor. They indicate where the receiver device will be situated in every iteration.

*Parking lot:* the marks are made each metre on the asphalt floor with chalk or white paint. For the 0.5 and 1.5 distances some marks are painted as well in order to obtain a greater accuracy since in principle in wireless communication the outside environments may have worse behaviour than the inside ones. This reason is also valid to explain why the step of the metric scale is each metre instead of each 2 metres as in the corridor.

*Grass field:* a rope as long as the desired metric scale (for example 25 metres) is used in this environment. The ends of the rope have to be tied to two nails which must be hammered in the grass in order to maintain the tightness of the rope. Other nails must not be utilized to make the distance marks because of their metallic material, which might have influence on the collected results. Therefore, adhesive tape must be used for that task. Finally, notice that the grass numeric scale is the same as parking lot one.

4. The performance of the experiment starts at this point and it is common for every scenario. The sender sensor on its box must be situated at the beginning of the metric scale, and the receiver one must be placed (on the same plane and with the same orientation as the sender) at the first tested distance. Activate the transmission of information and wait until the 20 packets have been sent. Then, move the receiver node to the next scale mark and activate the sending again. Perform this process until completing the whole metric scale.
5. Take out the collected data from the sensors and store it in another device memory.
6. Execute the phases 4 and 5 twice in order to be able to obtain a significant quantity of data and so our findings are more reliable.
7. Vary the transmission power value and complete the phases 4, 5 and 6.

Finally, three executions per transmission power value are achieved. The collected information of the three different tested variables must be studied to be able to state some conclusions. In this paper, this will be carried out in following sections.

## 2.2 Experiment B

Discovering the behaviour of the wireless sensor when its transmission angle with regard to the receiver is varied, is the main target of this experiment. For that, the reception rate feature is tested for every different wireless sensor orientation angle. In addition, received signal strength and LQI measurements can be also collected from the environment to accomplish a deeper analysis.

### 2.2.1 Tested parameter

Crucial decisions that have to be taken by the WSN's designer include how the different elements in the area must be placed as well as their orientation. In every communication system, several parameters should be optimised to reach the transmission and reception of information as successful as possible. One of these variables is the *angle of orientation* of each element in the network, it must be carefully studied in order to get the greatest system capabilities.

In most of the cases there are some directions in which a sender transmits with more power than others. This effect depends not only on the kind of the device's antenna but also on some other factors what make impossible to define its actual radiation diagram/pattern before being tested in practice. For this reason, although the theoretical behaviour of the wireless sensor's antenna was known, a detailed experiment must be executed on the real scenario where the WSN will be deployed in order to obtain a complete description of its behaviour.

### 2.2.2 Detailed experiment

The procedure carried out in this experiment is quite similar to the one executed previously, though two more devices are used to take measurements on the grass field scenario. Furthermore, practically the same software applications and instruments are utilized as well.

1. The Experiment A programs can be loaded in the wireless sensors of this experiment. In addition, the transmission power can have any value but it has to be the same for all the orientation angle executions. Nevertheless, the selected power should not be too low in order to collect a wide variety of measurements since for not very high powers the reach of the communication is quite poor mainly in outside environments.
2. Both sensors are fixed to the boxes which have the same dimensions as those used in the previous experiment. In the grass field case one transmitter and four received devices are utilized to take measurements and so five identical boxes are necessary.
3. The metric scales are created as in the Experiment A as well, except for the grass field case where two ropes of 50 metres of length with nails in the four ends must be hammered in the grass. The ropes, with their centres joined, must shape a cross of four right angles in the area. In addition, the scale marks have to be made in the four directions starting from the centres.
4. The experiment execution is different for the grass field and the rest of environments:

*Corridor/Parking lot:* transmitter and receiver must be situated at the beginning of the metric scale and on the first distance mark respectively, both with the same orientation angle, which is defined as 0 degrees.

*Grass field:* the sender is placed where both rope centres are joined, whereas the receivers are situated in the first point of the four scales. Every sensor must have the same orientation angle (0 degrees).

Later, the user has to activate the transmission of the 20 UDP packets and when they have been sent, the receivers must be moved forward. Repeat it until the end of the metric scales.

5. Extract the information from the wireless sensors' memory.
6. The phases 4 and 5 must be performed twice.
7. The way to test the communication between two nodes according to their orientation angles is also different for the grass field case and the other environments:

*Corridor/Parking lot:* the transmitter fixed to its box must be rotated 90 degrees and then the phases 4, 5 and 6 have to be completed. Rotate it 90 degrees again and repeat the process until the sender orientation angle is 360 degrees, it means until the transmitter reach its initial position (0 degrees). Later, rotate the receiver 90 degrees and repeat

the experiment while the sender device is rotated four times (0, 90, 180 and 270 degrees). Finally, the phases 4, 5 and 6 have to be executed sixteen times, being the devices' orientation different in each one. Figure 1 shows an example of communication with the position of the sender device at 0 degrees and the receiver one at 180 degrees.

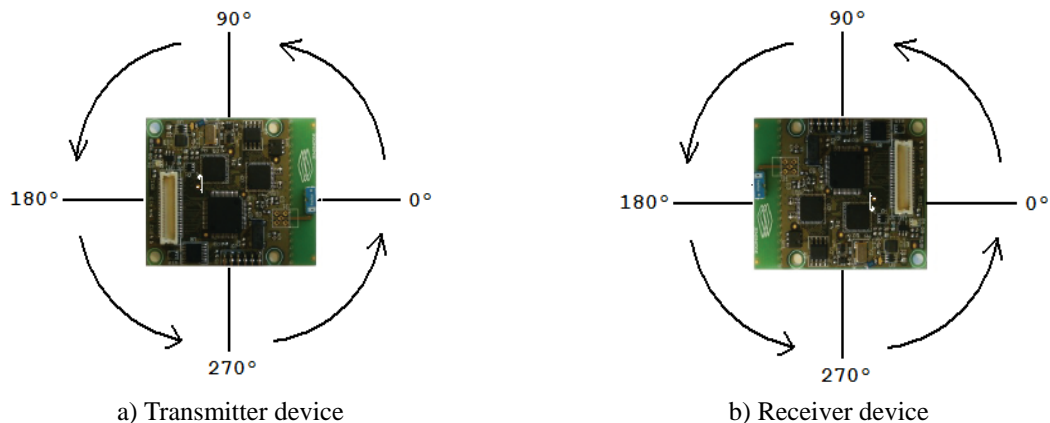


Figure 1. Example of communication between devices in the Experiment B

*Grass field:* the use of four devices to capture information make the orientation tests easier on this area. In contrast to the process carried out in the other environments, the receivers do not have to be rotated in the whole experiment. However, the sender device must be rotated 90 degrees when the phases 4, 5 and 6 have finished. Perform this process until four sets of measurements (one per transmitter orientation angle) have been collected.

Only a single receiver is utilized in both corridor and parking lot as their dimensions do not allow us to conduct a full experiment. On the other hand, in this wireless sensor communication, sixteen different positions between a sender and a receiver are tested for every environment. It lets the user obtain a wide definition of the sensors' behaviour when its orientation angle with regard to another device changes.

### 2.3 Experiment 3

The parameter tested in this experiment is the battery life. A summary of some variables which can have influence on the battery discharge is presented in the first subsection, whereas the systematic process to test the batteries of any wireless sensor is described in the second one.

#### 2.3.1 Tested parameter

Due to the wireless characteristic of the sensors and to their obligatory tiny size when they constitute a network, their batteries have serious restrictions in energy storage. As a result, the devices' lifetime and so the network's lifetime depends on how that energy is wasted. Therefore, possibly *battery life* is the most critical characteristic in a WSN and the creator has to be aware of this fact when designing so that the energy waste of the elements is as low as possible and every function of the system is carried out.

The dependence of the battery discharge on numerous variables can be discovered when the devices are tested on real scenarios. Some of these variables are studied in this experiment. Investigations as one carried out by Park, Lahiri and Raghunathan in [11] prove that the energy consumption in wireless sensors is influenced by factors such as sampling interval (time between two consecutive transmissions), ambient temperature, transmission time (total time that the device is transmitting per sampling interval) and transmission power level. The latter together with the number of sent and received messages are the variables utilized to study the sensor's battery life characteristic in this paper.

#### 2.3.2 Detailed experiment

Before testing the battery life of the devices according to the transmission power and the number of sent and received messages, an initial experiment without including these three variable, must be executed. The software application loaded in the sensor, to carry out this first experiment, must perform the same code as the sender and receiver applications in the main experiment execute, excluding the sending and receiving tasks. The battery measurements must be collected with the voltmeter every 15 minutes for the two first hours and every 30 minutes for the rest of the experiment, since the battery level decrease faster at the beginning. Therefore, the battery life characteristic will be defined quite better if the sampling

interval is shortened. After this initial experiment, the next process must be completed:

1. Load both transmitter and receiver program in the different wireless sensors. They must be programmed so that 10 packets per second are sent from a device to the other. Furthermore, while in the transmitter application a variable counts the sent messages, in the receiver the captured packets must be counted by another one. The device has to send the packets to the rate indicated previously for an indefinite time. In addition, a mechanism to change the transmission power has to be programmed.
2. Put the wireless sensor on different boxes, which have the same dimensions as in the other experiments, and place them in the corridor, just 0.5 metres separated each other.
3. Establish the level of transmission power and activate the sending of messages.
4. Measurements of the battery's voltage level must be collected every 15 minutes for the first two hours. After this time, they must be taken every 30 minutes. At the same time, the number of sent and received packets must be extracted from the devices memory.
5. This process has to be executed until both sensor batteries are discharged, it means until they can not carry out the transmission or reception task.
6. Perform the phases 4 and 5 as many times as transmission power levels are going to be tested. In every iteration, the batteries must be charged or replaced by other ones with the same features.

After executing the Experiment C, for every transmission power the user has a battery life performance which depends on the time as well as the sent and received messages. Also, he has a set of measurements that describe the behaviour of the battery level without sending and receiving information.

### **3. Results**

The previous technique for the instrumentation of Wireless Sensor Network nodes has been executed, and in this section, the results obtained from the three different experiments are presented. Specifically, the Micro.2420 U100 devices [12] of Sensinode company [13] have been tested. These sensors, with a size of 40 x 50 mm, have NanoStack architecture [14] and have support for running on 2 NiMH batteries. In addition, they have 4Mbit serial data FLASH memory to store information and an Fractus on board antenna [15].

#### **3.1 Experiment A**

Different relationships between the Micro.2420 U100 sensor's characteristics tested in this experiment and the distance, transmission power and environment are shown and analysed below.

##### **3.1.1 Reception rate**

Figure 2 shows the results of running the Experiment A for every environment. It can be observed that generally the higher the transmission power level, the greater both reception rate (reception probability) and maximum reach of the communication. Also, it can be seen that the reception rate presents very poor results. That is because just in the corridor the reception probability maintains a value near to 1 for long distances when the devices transmit with high power. The quality of the communication in the rest of environments is quite mediocre and so it makes difficult the deployment of a WSN in them.

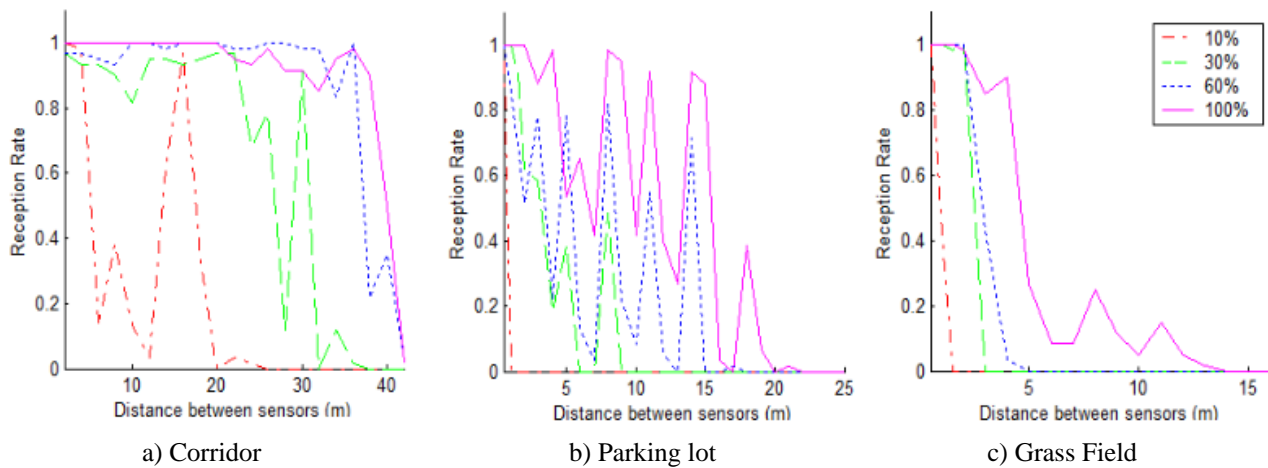


Figure 2. Reception rate vs distance for different transmission powers in every environment

The existence of zones where the communication is worse than in other nearer ones, are shown mainly in the parking lot graph. One of the causes of these irregularities might be the propagation phenomenon called multi-path, which occurs when the receiver's antenna captures the same signal from two or more paths. These signals might come reflected from different objects of the environment, such as cars and trees. This fact as well as the measurements in the grass field were collected without possible objects close of the sensors, prove the non existence of irregularities in the grass field graph. In the figure also can be appreciate easily that the maximum communication reach is obtained in the corridor. This fact could be directly related with the multi-path phenomenon as well. Because of this indoor placement is quite narrow, if it is compared with the outside environments, the receiver's antenna would capture stronger signals as well as a greater number of them which would come, after reflecting on the walls, from more different paths. Therefore, we can conclude that multi-path is a phenomenon which has more presence in those places where the reflections are more usual and it might help to capture easily the signal in the receiver's antenna and hence would improve the communication between devices.

When any telecommunication system is created, it must be made according to some quality criteria. The reception rate (or its complement, the packet loss rate) is one of them often. Normally, the creator must design the system so that all the elements of the communication maintain a probability of delivery higher than a concrete value. This fact could be easily achieved if the devices transmit with their greatest power, but then it could cause a quick discharge of their battery. It is responsibility of the WSN's creator choose the correct decisions. To give a practical example using the data collected from the Experiment A: it is desirable to plan a WSN in a corridor so that the battery life of the sensors is as long as possible and the reception rate is not less than 0.9. Observing Figure 2-a, for every transmission power, its maximum reach where the reception rate is equal or higher than 0.9, must be selected. Therefore, the network nodes could transmit with the 10% of their power just until 4 meters to maintain the quality criterion. With the 30% of the power the maximum reach is 22m, and with the 60% and the 100% this is 32m and 38m respectively.

### 3.1.2 Received signal strength

The results achieved after testing this parameter can be appreciated in the Figure 3. First of all, notice that the minimum value of this factor obtained in the experiment was -94 dBm. However, in the graph, values lower can be seen. This fact must not confuse to the reader since these values equivalent to 0 dBm. It has been made to give more clarity to the graph results. Observing the figure, in every environment it can be seen that the received signal strength by the sensor decreases when the transmission power diminishes and the distance between wireless sensors is increased. In addition, some irregularities in the communication exist in both corridor and parking lot environments mainly, possibly as a result of multi-path phenomenon. In addition, in Figure 3-d it can be appreciated that the corridor is the scenario where the devices receive better signal (these results are obtained for a transmission power level of 100%).

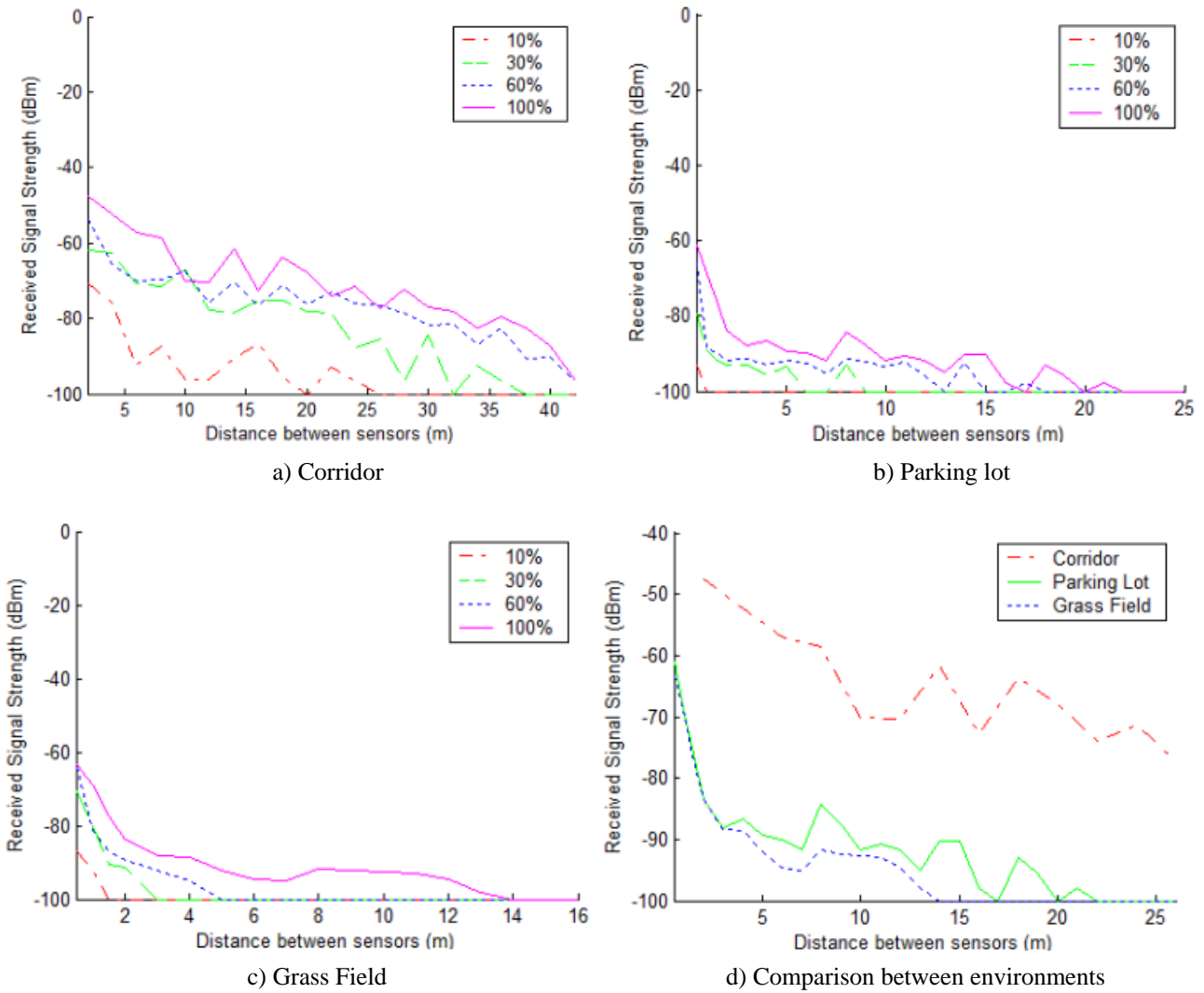


Figure 3. Received signal strength vs distance between sensors

### 3.1.3 Link quality indicator

For every environment the received signal strength by the sensor and LQI has the same dependence with regard to the distance between sensors and the transmission power. In Figure 4 it can be seen that LQI diminishes when the reach is higher. Furthermore, this indicator of quality decreases, and so the error rate is higher, if the transmission power is lower.

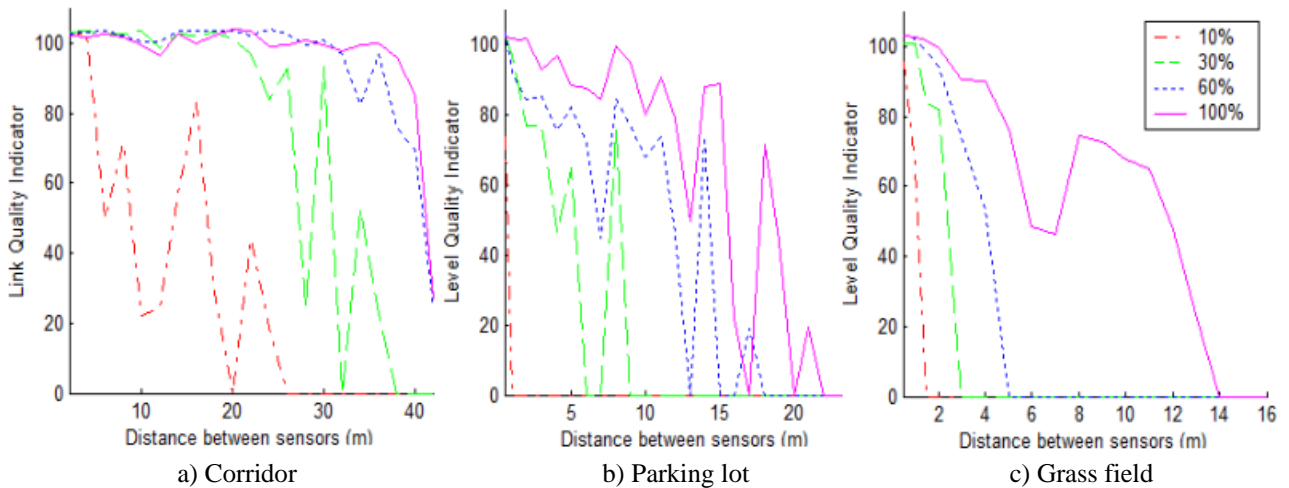


Figure 4. LQI vs distance for different transmission powers in every environment.

### 3.2 Experiment B

Figure 5 shows the results in the grass field scenario when the experiment is executed completely with the 100% of transmission power. There are four different graphs, one per transmitter sensor orientation. Furthermore, each graph describes the reception rate of the receiver sensors in their four different positions. The most remarkable aspect of this test is the finding of the suitable orientation angle of both sender and receiver sensors so that the communication has as high quality as possible. This happens when the transmitter's orientation is 180 degrees and the receiver's one is placed at 270 degrees with regard to the sender device (Figure 5-c). In these positions, the sender's antenna is transmitting with its maximum directivity towards that receiver, which is capturing information with its highest directivity as well (this can be seen in the radiation pattern of the Fractus antenna in the appendix).

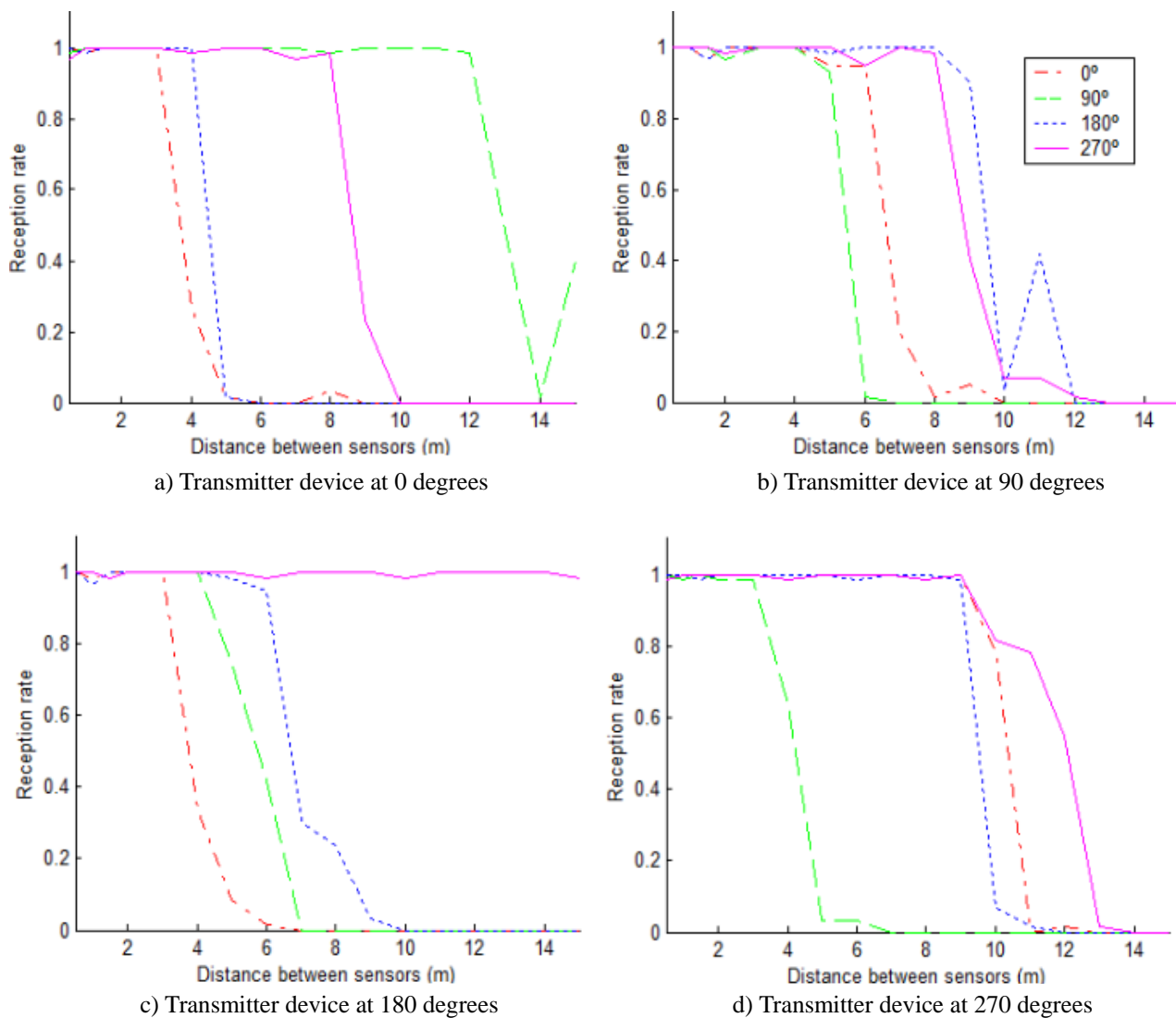


Figure 5. Reception rate vs distance for every transmitter orientation angle in the grass field

### 3.3 Experiment C

Figure 6 shows the ratio between the battery level and the received and transmitted messages for two different powers (6-a and 6-b). It can be observed that in both cases of transmission power, the battery level is discharged in the receiver in a shorter time than in the sender. In addition, for the 10% transmission power, the battery life of the wireless sensor is longer than for the 100%, since the device sends a higher number of packets. However, it is noticeable that the number of messages received with both powers is quite similar. That is logic because the transmission power just must affect to the sender. Therefore, it can be concluded that the battery life of a sensor is longer when the transmission power diminishes. Furthermore, in Figure 6-c it is seen that an isolated sensor saves more battery level than one which transmits or receives, even when it sends messages with the minimum power. The battery lives of the sensors in the experiments shown in the graphs a) and b) were 100 minutes approximately shorter than that shown in c).



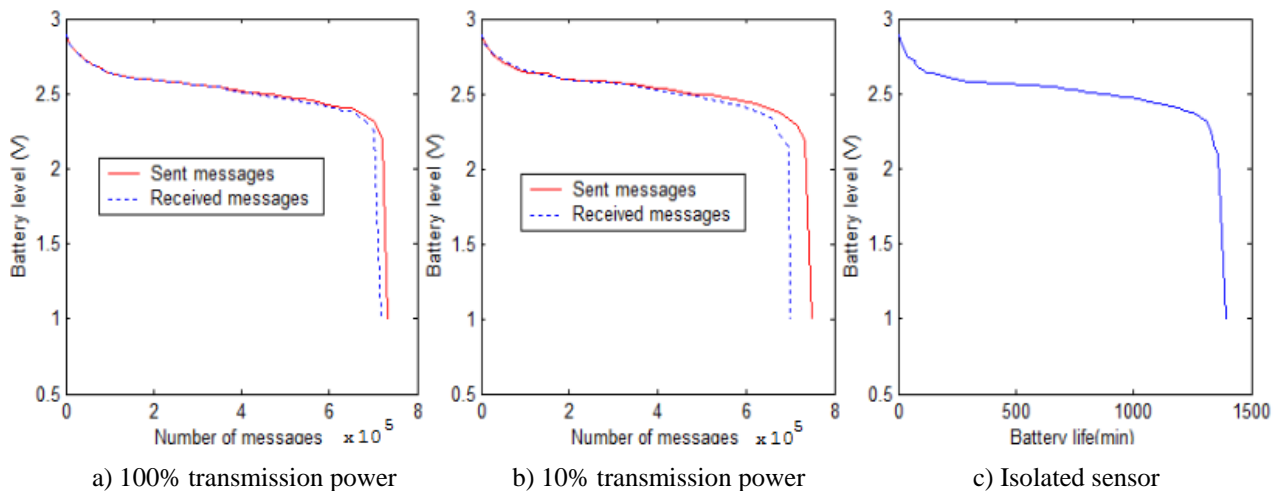


Figure 6. Battery level vs number of messages / battery life

#### 4. Conclusions

A systematic experimental procedure to achieve the instrumentation of any Wireless Sensor Network node has been described in this paper. This process is divided into three experiments. Firstly, characteristics of the device and the wireless communication such as the reception rate, the received signal strength and LQI are tested. Secondly, a simple experiment is executed to discover the nodes' behaviour when their orientation is varied. Finally, various tests about the battery life of the sensors are carried out in order to establish some conclusions concerning the influence of different factors on this.

This technique has been executed over a specific commercial wireless sensor. The analysed results show a great difference in the quality of the communication between the corridor scenario and the two outside environments. Also, it is noticeable the numerous irregularities which exist when the devices transmit and receive information. These might be caused by the multi-path phenomenon. In addition, it is demonstrated that the sensors' behaviour according their orientation is similar to that described by the theoretical antenna's radiation diagram. Finally, it is shown that an increase of the transmitted power affects on the discharge of battery, which is more influenced by receiving than transmitting.

#### 5. References

- [1] "First International Workshop Data Mining and Wireless Sensor Networks (DM-WSN)", Hong Kong, December 18 - 22, 2006, <http://cs.acadiau.ca/~shussain/dm-wsn/cfp.html>
- [2] Edgar H. Callaway Jr.: "Wireless sensor networks: architectures and protocols", pages 1 - 11, 2004
- [3] Jerry Zhao and Ramesh Govindan: "Understanding Packet Delivery Performance In Dense Wireless Sensor Networks", *SenSys'03 Conference*, November 5-7, 2003, Los Angeles, California, USA.
- [4] Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, Tian He, and John A. Sankovic: "ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks", *SenSys'06*, November 1-3, 2006, Boulder, Colorado, USA.
- [5] M. Horton, D. Culler, K. Pister, J. Hill, R. Szweczyk, and A. Woo: "MICA, The Commercialization of Microsensor Motes", in *Sensors Magazine*, pages 40–48, April 2002.
- [6] S. Blom, C. Bellettini, A. Sinigalliesi, L. Stabellini, M. Rossi, and G. Mazzini: "Transmission Power Measurements for Wireless Sensor Nodes and their Relationship to the Battery Level", 2005.
- [7] Eyes Project: "EYES: Energy Efficient Wireless Sensor Networks", <http://www.eyes.eu.org/>
- [8] J. Postel: "User Datagram Protocol", RFC 768, 28 August 1980
- [9] "Transmission Control Protocol", RFC 793, September 1981
- [10] IEEE 802.15.4, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)", 1999. IEEE Std. 802.15.4, 2003.
- [11] C. Park, K. Lahiri, and A. Raghunathan: "Battery Discharge Characteristics of Wireless Sensor Nodes: An Experimental Analysis", 2005.
- [12] Sensinode Ltd.: "U100 Micro.2420 Flyer", 2007.
- [13] Sensinode Ltd., [www.sensinode.com](http://www.sensinode.com)
- [14] Sensinode Ltd.: "NanoStack Reference v1.0.0", 2007.
- [15] FRACTUS, S.A.: "Slim Reach Xtend™ Antenna Datasheet", 2007, [www.fractus.com](http://www.fractus.com)

**Appendix:** Fractus antenna radiation pattern

