



# ANÁLISIS TOPOLÓGICO DE REDES DE SENSORES INALÁMBRICOS EN ENTORNOS INTERIORES COMPLEJOS

***Peio Lopez Iturri***

*Francisco Falcone Lanas*

*Carlos Fernández-Valdivielso*



## AGRADECIMIENTOS

Desde aquí me gustaría agradecer el apoyo, la ayuda, el compañerismo y la amistad recibida todos estos años por todos los compañeros y compañeras con los que he coincidido mientras estudiaba en la Universidad Pública de Navarra.

También me gustaría agradecer la labor de todos los docentes que en algún momento de la carrera han sido mis profesores, ya que aunque de unos se aprende más que de otros, siempre se aprende.

Gracias también a Francisco Falcone y a Carlos Fernández-Valdivielso, que han ejercido de tutores en este proyecto fin de carrera, y sin los que no habría sido posible realizarlo. Gracias por vuestros consejos, ayuda y, sobre todo, por todo lo que he podido aprender.

Un agradecimiento especial para Juan Antonio Nazabal, con el que he compartido muchas horas de duro trabajo en los laboratorios de la Universidad Pública de Navarra.

Y por supuesto, agradecer a mi madre y a mi padre el sacrificio hecho, por el que este día ha podido llegar.

Y a Eva, por estar siempre ahí.

Eskerrik asko denoi!

¡Muchas gracias a tod@s!



# ÍNDICE

<b>Listado de figuras y tablas</b> .....	6
<b>1.-Introducción</b> .....	10
<b>2.-Objetivo del proyecto</b> .....	11
<b>3.-Estado del arte</b> .....	12
<b>4.-Propagación radioelétrica</b> .....	15
4.1.- Pérdidas de propagación.....	15
4.2.- Penetración de la señal. Absorción .....	16
4.3.- Reflexión, Dispersión y Difracción.....	19
4.4.- Propagación multitrayecto .....	22
4.4.1.-Desvanecimiento multitrayecto .....	23
4.4.2.-Modelos de propagación en interiores .....	25
4.4.2.1-Modelos de propagación deterministas.....	26
4.4.2.2-Modelos basados en técnicas de trazado de rayos (Ray Tracing) .....	26
<b>5.-Redes de sensores inalámbricos. ZigBee</b> .....	28
5.1.- Redes de sensores.....	28
5.1.1.-Introducción .....	28
5.1.2.-Redes de sensores inalámbricos .....	29
5.1.2.1-Estándares para una WSN.....	31
5.1.2.2.-Características clave y restricciones en el diseño de WSNs .....	31
5.2.- ZigBee .....	35
5.2.2.-Características de ZigBee.....	36
5.2.2.1.-Alta fiabilidad (Highly reliable).....	36
5.2.2.2.-Bajo coste .....	38
5.2.2.3.-Bajo consumo de energía .....	38
5.2.2.4.-Altamente seguro.....	39
5.2.2.5.-Estándar abierto.....	39
5.2.2.6.-Baja tasa de transmisión de datos.....	39
5.2.3.-ZigBee y 802.15.4 .....	40
5.2.4.-Frecuencias y velocidades de transmisión .....	42
5.2.5.-Interoperabilidad .....	46
5.2.6.-Tipos de dispositivos.....	46
5.2.7.-Roles de los dispositivos .....	46
5.2.8.-Topologías de red.....	49
5.2.9.-Aspectos básicos de la comunicación .....	50
5.2.9.1.-CSMA-CA.....	51
5.2.9.2.-Beacon-Enabled frente a Nonbeacon Networking.....	51
5.2.9.3.-Métodos de transmisión de datos .....	52
5.2.9.4.-Verificación de los datos .....	54



5.2.9.5.-Direccionamiento .....	54
5.2.10.-Asociación y disociación .....	55
5.2.11.-Vinculación de dispositivos (Binding).....	55
5.2.12.-Funciones y estructura de trama de las distintas capas del estándar IEEE	
802.15.4 y ZigBee .....	56
5.2.12.1.-Capa física (PHY) .....	56
5.2.12.1.1-Estructura general del paquete PHY: .....	57
5.2.12.2.-Capa MAC .....	58
5.2.12.2.1.-Trama piloto: .....	58
5.2.12.2.2.-Trama de datos: .....	59
5.2.12.2.3.-Trama ACK: .....	59
5.2.12.2.4.-Trama de comando MAC: .....	60
5.2.12.3.-Capa de red (NWK) .....	60
5.2.12.4.-Capa de aplicación .....	60
5.2.12.5.-Seguridad.....	61
5.2.13.-La pasarela ZigBee.....	61
5.2.14.-ZigBee frente a otras tecnologías inalámbricas.....	62
<b>6.-Dispositivos inalámbricos utilizados en el estudio.....</b>	<b>66</b>
6.1.- XBee-Pro.....	66
6.2.- Dispositivos de la Universidad de Valencia.....	72
<b>7.-Resultados obtenidos.....</b>	<b>76</b>
7.1.-Métodos para la obtención de los datos .....	76
7.1.1.-Potencia recibida (RSSI) .....	76
7.1.1.1-Analizadores de espectro.....	76
7.1.1.2-Dispositivos ZigBee .....	78
7.1.1.3-Simulación con trazado de rayos.....	80
7.1.2.-Calidad (PER) .....	82
7.1.3.-Consumo .....	84
7.2.-Resultados obtenidos.....	85
7.2.1.-Escenario 1: RSSI (Received Signal Strength Indicator).....	85
7.2.2.-Escenario 2: RSSI + Consumo con varios transmisores .....	91
7.2.3.-Escenario 3: RSSI con diferentes obstáculos .....	98
7.2.4.-Escenario 4: Primeras medidas de PER .....	100
7.2.5.-Escenario 5: Mejoras en medidas de PER.....	102
7.2.6.-Escenario 6: PER con 1.000.000 paquetes transmitidos .....	105
7.2.7.-Escenario 7: RSSI, PER y trazado de rayos .....	109
7.2.8.-Escenario 8: Dispositivos CyFi. Universidad de Valencia .....	111
7.2.9.-Escenario 9: Interferencia WiFi .....	119
7.2.10.-Escenario 10: Interferencia de un microondas .....	121
7.2.11.-Escenario 11: Movilidad .....	123
<b>8.-Conclusiones .....</b>	<b>128</b>
<b>9.-Líneas futuras .....</b>	<b>131</b>
<b>Bibliografía.....</b>	<b>132</b>



---

<b>ANEXOS</b> .....	134
A.- Congresos: .....	134
B.- Código de los programas Java: .....	141
B.1.- ZigBeePERSender: .....	142
B.2.- ZigBeePERReceiver: .....	150
B.3.- ZigBeePERReceiverV: .....	159
B.4.- Gráficas (Evolución temporal): .....	167
C.- Sniffer Atmel: .....	175
D.- Datos obtenidos en los escenarios: .....	177



## Listado de figuras y tablas

<b>Tabla 3.1.</b> Precios de software + módulos ZigBee. ....	12
<b>Figura 3.1.</b> Empresas con soluciones ZigBee. ....	13
<b>Tabla 4.1.</b> Adquisición de los datos y actuación en un sensor. ....	16
<b>Figura 4.1.</b> Medida de la penetración de señal en una pared de yeso. Figura obtenida de [2]. ....	17
<b>Figura 4.2.</b> Potencia de señal recibida para los ángulos de transmisión mostrados en la figura 4.1. Figura obtenida de [2]. ....	17
<b>Figura 4.2.</b> Ley de Snell, refracción. ....	18
<b>Figura 4.3.</b> Ley de Snell, reflexión. ....	19
<b>Figura 4.4.</b> Esquema del fenómeno de dispersión. ....	20
<b>Figura 4.5.</b> Esquema del fenómeno de difracción. ....	20
<b>Figura 4.6.</b> La señal puede llegar al receptor de varias formas: (1) Penetración, (2) difracción, (3) reflexión y (4) dispersión. Figura obtenida de [2]. ....	21
<b>Figura 4.7.</b> Escenario estático multitrayecto. Figura obtenida de [2]. ....	22
<b>Figura 4.8.</b> Suma de fases, (a) en fase, (b) en contrafase. ....	23
<b>Figura 4.9.</b> Desvanecimiento. Figura obtenida de [5]. ....	24
<b>Tabla 4.1.</b> Características de los desvanecimientos multitrayecto. Tabla obtenida de [5]. ....	25
<b>Figura 4.10.</b> Ecuaciones de Maxwell. ....	26
<b>Figura 4.11.</b> Método de <i>Ray Launching</i> . ....	27
<b>Figura 5.1.</b> Adquisición de los datos y actuación en un sensor. Figura obtenida de [3]. ....	29
<b>Figura 5.2.</b> Redes de sensores inalámbricos. Figura obtenida de [3]. ....	30
<b>Tabla 5.1.</b> Comparación de las redes tradicionales y las WSN. Figura obtenida de [3]. ....	34
<b>Figura 5.3.</b> Miembros de The ZigBee Alliance. Figura obtenida de [1]. ....	35
<b>Figura 5.4.</b> ZigBee y el estándar IEEE 802.15.4. ....	36
<b>Figura 5.5.</b> Mesh Networking con ruta establecida. Figura obtenida de [1]. ....	37
<b>Figura 5.6.</b> Mesh Networking con ruta alternativa. Figura obtenida de [1]. ....	37
<b>Figura 5.7.</b> Tiempo de vida de la batería (AA) en ZigBee. Figura obtenida de [1]. ....	38
<b>Figura 5.8.</b> Capas del protocolo de red de ZigBee. Figura obtenida de [2]. ....	41
<b>Tabla 5.2.</b> Frecuencias y velocidad de transmisión en IEEE 802.15.4. Figura obtenida de [2]. ....	42
<b>Figura 5.9.</b> Canales IEEE 802.15.4. Figura obtenida de [6]. ....	43
<b>Figura 5.10.</b> Modulación BPSK. ....	44
<b>Figura 5.11.</b> Modulación ASK. ....	44
<b>Figura 5.12.</b> Modulación O-QPSK. ....	45
<b>Figura 5.13.</b> DSSS y PSSS ayudan a mejorar la SNR (a) y a reducir el efecto de interferencias (b). Figura obtenida de [2]. ....	45
<b>Figura 5.14.</b> Roles de los dispositivos según el estándar ZigBee y el estándar IEEE 802.15.4. Figura obtenida de [2]. ....	47
<b>Figura 5.15.</b> Tipos de nodos ZigBee. ....	48
<b>Figura 5.16.</b> Asociación en una red de los distintos tipos de nodos ZigBee. ....	48
<b>Figura 5.17.</b> Topología de red en estrella. Figura obtenida de [2]. ....	49
<b>Figura 5.18.</b> Topología de red peer-to-peer (en malla). Figura obtenida de [2]. ....	49

<b>Figura 5.19.</b> Topología en árbol de ZigBee. Figura obtenida de [2].	50
<b>Figura 5.20.</b> Transmisión de datos a un coordinador en IEEE 802.15.4: (a) con trama piloto o beacon, y (b) sin trama piloto. Figura obtenida de [2].	52
<b>Figura 5.21.</b> Transmisión de datos de un coordinador en IEEE 802.15.4: (a) con trama piloto o beacon, y (b) sin trama piloto. Figura obtenida de [2].	53
<b>Figura 5.22.</b> Mediante el <i>bindig</i> se pueden vincular lógicamente dos o más dispositivos ZigBee. Figura obtenida de [1].	56
<b>Figura 5.23.</b> ZigBee y el estándar IEEE 802.15.4	57
<b>Figura 5.24.</b> Estructura de paquete ZigBee. Figura obtenida de [2].	57
<b>Figura 5.25.</b> Estructura de trama piloto de la capa MAC. Figura obtenida de [2].	58
<b>Figura 5.26.</b> Estructura de trama de datos de la capa MAC. Figura obtenida de [2].	59
<b>Figura 5.27.</b> Estructura de trama ACK de la capa MAC. Figura obtenida de [2].	59
<b>Figura 5.28.</b> Estructura de trama de comando de la capa MAC. Figura obtenida de [2].	60
<b>Figura 5.29.</b> Pasarela de ZigBee. Figura obtenida de [1].	62
<b>Figura 5.30.</b> Clasificación de redde inalámbricas de corto alcance. Figura obtenida de [2].	63
<b>Figura 5.31.</b> Comparación de tecnologías inalámbricas. Figura obtenida de [1].	64
<b>Figura 5.32.</b> Comparación de ZigBee con Bluetooth y 802.11b. Figura obtenida de [2].	64
<b>Tabla 5.3.</b> Comparación de ZigBee con Bluetooth y 802.11b.	65
<b>Figura 5.33.</b> Comparación de ZigBee con Bluetooth y 802.11b.	65
<b>Figura 6.1.</b> Módulo XBee-Pro.	66
<b>Figura 6.2.</b> Software X-CTU.	67
<b>Tabla 6.1.</b> Especificaciones del módulo Xbee-Pro. Tabla obtenida de [11].	68
<b>Figura 6.3.</b> Diagrama de radiación de la antena dipolo del módulo Xbee-Pro. Figura obtenida de [11].	69
<b>Figura 6.4.</b> Diagrama de radiación de la antena whip-monopolo del módulo Xbee-Pro. Figura obtenida de [11].	69
<b>Figura 6.5.</b> Diagrama de radiación de la antena chip del módulo Xbee-Pro. Figura obtenida de [11].	70
<b>Figura 6.6.</b> Dispositivo XBee-Pro alimentado por baterías.	70
<b>Figura 6.7.</b> Lectura de RSSI en XBee-Pro.	71
<b>Figura 6.8.</b> Velocidades de transmisión estándares para XBee-Pro.	71
<b>Figura 6.9.</b> Prototipo de la Mota CyFi de la Universidad de Valencia.	72
<b>Figura 6.10.</b> Mota CyFi de la Universidad de Valencia.	72
<b>Figura 6.11.</b> Software para los sensores de las motas CyFi.	73
<b>Figura 6.12.</b> Software para el RSSI y nivel de potencia transmitida de las motas CyFi.	73
<b>Figura 6.13.</b> Características del transceiver CyFi CYRF7936 de Cypress. Figura obtenida de [13].	74
<b>Figura 6.14.</b> Antena usada por las motas.	75
<b>Figura 6.15.</b> Diagramas de radiación para la antena de las motas. Figura obtenida de [14].	75
<b>Figura 7.1.</b> Analizador de espectros Agilent FieldFox N9912A.	77
<b>Figura 7.2.</b> Antena OAN-1070 de Level One. Figura obtenida de [15].	77
<b>Figura 7.3.</b> Diagramas de radiación de la antena OAN-1070 de Level One. Figura obtenida de [15].	78
<b>Figura 7.4.</b> Montaje PC + dispositivos ZigBee. Figura obtenida de [1].	78



<b>Figura 7.5.</b> Programa en Java para la lectura de RSSI.....	79
<b>Figura 7.6.</b> Elección del puerto USB y velocidad de transmisión.....	79
<b>Figura 7.7.</b> Programa para el envío de RSSI.....	80
<b>Figura 7.8.</b> Elección de dirección de destino.....	80
<b>Figura 7.9.</b> Escenario interior en 3D con los rayos lanzados desde un transmisor situado en el centro.....	81
<b>Figura 7.10.</b> Mapa de potencia recibida mediante trazado de rayos 3D, a una altura de 0.8 metros, con un único transmisor situado en el centro.....	81
<b>Figura 7.11.</b> Programa en Java para la lectura del PER.....	82
<b>Figura 7.12.</b> Programa en Java para el envío de paquetes en el caso del PER.....	83
<b>Figura 7.13.</b> Valores de número de paquetes a transmitir el caso del PER.....	83
<b>Figura 7.14.</b> Osciloscopio Tektronix DPO 3014.....	84
<b>Figura 7.15.</b> Ley de Ohm.....	84
<b>Figura 7.16.</b> Plano del laboratorio L5.....	85
<b>Figura 7.17.</b> Plano del despacho D5.....	86
<b>Figura 7.18.</b> Fotografía del laboratorio L5.....	86
<b>Tabla 7.1.</b> Distribución de potencia en el laboratorio L5.....	87
<b>Tabla 7.2.</b> Distribución de potencia en el despacho D5.....	87
<b>Figura 7.19.</b> Cálculo con modelos de propagación.....	88
<b>Tabla 7.3.</b> Valores para Cost-231, Keenan-Motley y Multi-Wall para L5.....	88
<b>Tabla 7.4.</b> Valores del modelo ITU-R P.1238 para el laboratorio L5.....	89
<b>Tabla 7.5.</b> Linear Path Atenuation Model para el laboratorio L5.....	89
<b>Tabla 7.6.</b> Modelo Cost-231 para el despacho D5.....	89
<b>Tabla 7.7.</b> Modelo Keenan-Motley para el despacho D5.....	90
<b>Tabla 7.8.</b> Modelo Multi-Wall para el despacho D5.....	90
<b>Tabla 7.9.</b> Linear Path Atenuation Model para el despacho D5.....	90
<b>Tabla 7.10.</b> ITU-R P.1238 para el despacho D5.....	90
<b>Figura 7.20.</b> Localización de los dispositivos para el escenario 2.....	92
<b>Figura 7.21.</b> Consumo XBee-Pro a nivel 4.....	93
<b>Figura 7.22.</b> Consumo XBee-Pro a nivel 3.....	94
<b>Figura 7.23.</b> Consumo XBee-Pro a nivel 2.....	94
<b>Figura 7.24.</b> Consumo XBee-Pro a nivel 1.....	95
<b>Figura 7.25.</b> Consumo XBee-Pro a nivel 0.....	96
<b>Figura 7.26.</b> Comparación del consumo según el nivel de potencia transmitido.....	97
<b>Tabla 7.11.</b> Valores de PER para 100.000 paquetes transmitidos.....	101
<b>Figura 7.27.</b> Representación de PER para 100.000 paquetes transmitidos.....	101
<b>Tabla 7.12.</b> Valores de PER para 500.000 paquetes transmitidos.....	102
<b>Figura 7.28.</b> Representación de PER para 500.000 paquetes transmitidos.....	102
<b>Tabla 7.13.</b> Valores de PER para 500.000 paquetes transmitidos con ACK.....	103
<b>Figura 7.29.</b> Representación de PER para 500.000 paquetes transmitidos con ACK.....	103
<b>Figura 7.30.</b> Comparación de PER para 500.000 paquetes transmitidos sin ACK y con un ACK.....	104
<b>Figura 7.31.</b> Plano de la primera planta con posiciones para PER de 1.000.000 paquetes transmitidos.....	105
<b>Figura 7.32.</b> Plano de la planta baja con posiciones para PER de 1.000.000 paquetes transmitidos.....	106
<b>Tabla 7.14.</b> Valores de PER para 1.000.000 paquetes transmitidos.....	106
<b>Figura 7.33.</b> Representación de PER para 1.000.000 paquetes transmitidos.....	107
<b>Figura 7.34.</b> Evolución temporal de RSSI y paquetes recibidos.....	108





<b>Figura 7.35.</b> Laboratorio de Radiocomunicación, UPNA.....	109
<b>Figura 7.36.</b> Esquemático del trazado de rayos 3D del laboratorio de Radiocomunicación de la UPNA.....	109
<b>Tabla 7.15.</b> Valores de PER para 500.000 paquetes transmitidos con ACK.....	110
<b>Figura 7.37.</b> Representación de los datos de la tabla 7.15.....	110
<b>Figura 7.38.</b> Plano del laboratorio de Radiocomunicación de la UPNA con las posiciones de las medidas.....	111
<b>Tabla 7.16.</b> Valores de RSSI con motas CyFi.....	112
<b>Tabla 7.17.</b> Valores de RSSI con motas CyFi a máxima tasa de transmisión.....	113
<b>Figura 7.39.</b> Software 'ZigBeePERReceiverV'.....	113
<b>Figura 7.40.</b> Resultados de simulación (RSSI) del trazado de rayos 3D para el plano bidimensional a 0.81m de altura.....	114
<b>Figura 7.41.</b> Resultados de simulación (RSSI) del trazado de rayos 3D para el plano bidimensional a 2.3m de altura.....	114
<b>Figura 7.42.</b> Resultados de simulación para las alturas 0.81m y 2.3m, a lo largo del eje x del escenario interior analizado.....	115
<b>Figura 7.43.</b> Plano del escenario con las cuatro posiciones para la medida del consumo de las motas CyFi.....	116
<b>Figura 7.44.</b> Variación en el consumo de potencia de las motas en función de la posición. La curva inferior (13.2 mA) es para el modo en standby. El resto de las curvas van desde la posición más cercana a la más lejana.....	117
<b>Figura 7.45.</b> Detalle del consumo de potencia de la mota en el escenario interior. Los valores medios de las corrientes son proporcionales a la separación entre motas.....	117
<b>Figura 7.46.</b> Espectro de las motas CyFi.....	119
<b>Figura 7.47.</b> Espectro de las motas CyFi, a la izquierda. Espectro de la red WiFi de la UPNA, a la derecha.....	120
<b>Figura 7.48.</b> Espectro de los dispositivos XBee-Pro transmitiendo a máxima potencia.....	121
<b>Figura 7.49.</b> Espectro de un microondas común.....	121
<b>Figura 7.50.</b> Espectro del XBee-Pro, en el canal 12 del estándar ZigBee (pico de la izquierda). Espectro de un microondas común, a la derecha.....	122
<b>Figura 7.51.</b> Detalle del dispositivo transmisor.....	123
<b>Figura 7.52.</b> Imagen de satélite del escenario.....	124
<b>Figura 7.53.</b> Vista del origen de la trayectoria recorrida por el coche desde la posición del receptor.....	124
<b>Figura 7.54.</b> Vista del final de la trayectoria recorrida por el coche desde la posición del receptor.....	125
<b>Figura 7.55.</b> Número de paquetes recibidos y perdidos en función de la velocidad del coche.....	126
<b>Figura 7.56.</b> Evolución temporal de RSSI de los paquetes recibidos.....	126
<b>Figura B.1.</b> Captura de pantalla del programa NetBeans 7.0.....	141
<b>Figura C.1.</b> Kit RZRAVEN de Atmel.....	175
<b>Figura C.2.</b> Dispositivo RZRAVEN.....	175
<b>Figura C.3.</b> Software AVR Studio 4.....	176



---

# 1.-Introducción

Los rápidos avances en las áreas de diseño de sensores, tecnologías de la información y redes inalámbricas han creado las condiciones necesarias para que las redes de sensores inalámbricos proliferen. De hecho, el uso de redes de sensores inalámbricos está creciendo rápidamente en un gran número de campos de aplicación, como son la monitorización industrial, monitorización de habitats, agricultura y ganadería, monitorización estructural, domótica, asistencia sanitaria, localización y seguimiento o seguridad y defensa, entre otros muchos.

Por tanto, el estudio, análisis y desarrollo de esta tecnología en auge se presenta como algo fundamental de cara al futuro de las telecomunicaciones, y de la sociedad en su conjunto.

Siendo un Ingeniero de Telecomunicación la persona ideal para este desempeño, ha sido un placer para mí haber estado involucrado en este trabajo, aportando todo lo que ha estado en mi mano para seguir dando pasos hacia el futuro.



## 2.-Objetivo del proyecto

El objetivo de este proyecto fin de carrera es analizar el impacto de entornos interiores complejos en el despliegue de infraestructuras de redes de sensores inalámbricos. Concretamente, la influencia en los valores de RSSI, consumo de energía y PER (Packet Error Ratio) serán los datos más significativos en el estudio.

Se analizará especialmente la influencia de la topología y morfología de la red en varios escenarios interiores complejos mediante simulaciones basadas en el método de trazado de rayos 3D y mediante medidas reales hechas directamente con los dispositivos inalámbricos ubicados en dichos escenarios.

Teniendo en cuenta el rápido incremento de las aplicaciones en las que estas redes son esenciales (monitorización de habitats, monitorización de infraestructuras, asistencia sanitaria, domótica, defensa, agricultura, ganadería, etc.), entender su comportamiento en un escenario real es imprescindible para un adecuado y eficiente despliegue de las mismas.

Todas las redes y dispositivos analizados en este trabajo están basados en la norma IEEE 802.15.4. Esta norma, en la que el estándar basado en ella más conocido y utilizado es ZigBee, trata sobre las LR-PANs (Low Rate Personal Area Network), las cuales se caracterizan por estar los dispositivos alimentados por baterías. Por lo que encontrar la mejor configuración y distribución espacial de los dispositivos que forman la red de sensores, de tal forma que el consumo de energía sea el menor posible, será uno de los objetivos más importantes, ya que esto significaría alargar al máximo la vida de las baterías. Más teniendo en cuenta que las redes de sensores pueden contener varios centenares o millares de dispositivos. De igual forma, se mejoraría el impacto medioambiental.



## 3.-Estado del arte

Desde hace ya varios años, la tecnología ZigBee está al alcance de quien la necesite. Hoy en día existe una amplia y económicamente viable gama de dispositivos basados en el estándar ZigBee, y debido a ello, cada vez son más las aplicaciones en las que esta tecnología está presente.

La lista de empresas que ofrecen soluciones basadas en los estándares 802.15.4 y ZigBee es muy larga, por lo que a continuación se nombrarán las más importantes junto a alguno de los productos que ofrecen:

- **MeshNetics:** ZigBit 900 (868/915 MHz); ZigBit B0, ZigBit A2 y ZigBit Amp para la banda 2.4 GHz. [16]
- **Atmel:** ZigBit B0 y ZigBit A2. [17]
- **LS-Research:** ProFLEX01. [18]
- **Cirronet:** ZMN 2405-X y ZMN 2430-X. [19]
- **Panasonic:** PAN4555, PAN4561, PAN4566 y PAN 4570. [20]
- **Digi:** XBee y XBee-Pro. [21]
- **STMicroelectronics:** SPMB250-A1. [22]
- **OKI:** ML7222. [23]

Empresa	Producto	Coste unidad (\$)
Digi	XBee-Pro	<b>29,90</b>
Atmel	ZigBit B0/A2	<b>37,70</b>
LS-Research	ProFLEX01	<b>31,78</b>
Cirronet	ZMN 2405/ZMN 2430	<b>21,50 a 37</b>
<b>Panasonic</b>	<b>PAN4561</b>	<b>1: 58,06 → 250: 24,84</b>

**Tabla 3.1.** Precios de software + módulos ZigBee.

Los precios que aparecen en la tabla están referidos a la compra de una unidad. Si la compra es mayor, el precio se va reduciendo considerablemente, como puede verse en algunos casos, como Panasonic.

También existe la posibilidad de comprar únicamente el transceiver o el chip. Entonces, los precios varían entre 2 y 6 euros la unidad, y si el número de unidades es elevado, el precio puede reducirse hasta menos de 1 euro/unidad. Texas Instruments, Ember, Freescale, tienen una gran variedad de chips y transceivers a estos precios.

En la figura 3.1 se muestran varios productos de las empresas mencionadas.



Figura 3.1. Empresas con soluciones ZigBee.

También hay que destacar que actualmente en España existen varias empresas dedicadas a la implantación de redes de sensores inalámbricos. Algunas de ellas son empresas jóvenes, lo que indica que es un sector en alza. A continuación se nombran algunas de ellas:

- ❖ **WSVAL:** WSNVAL es una empresa situada en Valencia especializada en el diseño, desarrollo e implantación de aplicaciones basadas en redes de sensores inalámbricos. Ofrecen productos y soluciones a medida. Cuentan con 20 años de experiencia en sectores como medioambiente, agricultura, transporte, energía, industria y consumo, medicina, seguridad y domótica. [www.wsnval.com](http://www.wsnval.com)
- ❖ **SAYME:** Es una joven empresa española que centra su actividad en el diseño, desarrollo y fabricación de tecnología electrónica de redes de sensores inalámbricos de ultra bajo consumo. [www.sayme.es](http://www.sayme.es)
- ❖ **NEBUSENS:** Es una empresa dedicada a la investigación, desarrollo y comercialización de productos y soluciones tecnológicas innovadoras relacionadas con las redes inalámbricas de sensores, aplicables a sectores como el sanitario, doméstico o el industrial, entre otros. [www.nebusens.com](http://www.nebusens.com)



Hay que destacar también la implicación de un órgano como es el COIT (Colegio Oficial de Ingenieros de Telecomunicación) en el desarrollo y el apoyo de estas tecnologías. El año pasado, 2010, a fin de favorecer que los ingenieros de telecomunicación se impliquen en proyectos, desarrollos e implantaciones de redes de sensores en múltiples aplicaciones, el COIT organizó un ecosistema inicial de empresas especializadas en redes de sensores (que agrupa a empresas de tecnología, aplicaciones y a los operadores de telecomunicaciones) el cual pretende ser un escaparate de compañías y profesionales que comercialicen estas aplicaciones, mostrando que existe mercado y desarrollos tecnológicos que lo satisfacen.



## 4.-Propagación radioelétrica

Toda comunicación inalámbrica acarrea consigo poca fiabilidad debido a que el canal por donde se transmiten los mensajes, la atmósfera, carece de las fronteras físicas que aportan las comunicaciones por cable.

Existen varios factores que afectan a las comunicaciones inalámbricas, como el ruido de radiofrecuencia que producen las máquinas, cambios físicos en el entorno, las personas, los metales, cambios en la atmósfera, etc. Además, en la banda de frecuencias ISM de 2.4 GHz, que es la banda en que se centra este trabajo, existe mayor cantidad de interferencias. Todo ello podría causar que el envío de paquetes fallara.

Los sistemas de telecomunicaciones se diseñan para que la señal que llega al receptor cumpla una mínima relación señal a ruido que garantice la correcta recepción del mensaje transmitido. En el caso de las comunicaciones inalámbricas, las ondas electromagnéticas que llevan la información están expuestas a ciertos fenómenos que pueden alterar su propagación, influyendo en la potencia recibida, y por tanto, en la relación señal a ruido.

Por tanto, para la correcta planificación de un sistema de comunicación inalámbrico, como son las redes de sensores ZigBee, resulta esencial conocer los factores que pueden alterar la propagación electromagnética en la banda de frecuencia correspondiente. Por lo que entender las características de propagación de un entorno es esencial para evitar problemas y para una correcta implementación de la red.

En este apartado se van a explicar cuáles son estos efectos que se dan en la transmisión inalámbrica.

### 4.1.- Pérdidas de propagación

Al igual que todos los medios de transmisión, el aire también introduce una atenuación a la señal. Esta atenuación variará en función de la frecuencia de transmisión y de la distancia física existente entre emisor y receptor.

Estas pérdidas se pueden aproximar con la fórmula de Friis, que da una relación entre la potencia transmitida y la recibida:

$$\frac{P_r}{P_t} = \frac{1}{4 \cdot \pi \cdot r^2} \cdot D_T \cdot A_{efR} = \left( \frac{\lambda}{4 \cdot \pi \cdot r} \right)^2 \cdot D_T \cdot D_R = \left( \frac{1}{\lambda \cdot r} \right)^2 \cdot A_{efT} \cdot A_{efR}$$



En este trabajo, la antena transmisora y la antena receptora son iguales, y siempre se va a trabajar a la misma frecuencia. Por tanto, la deducción más importante que se saca de esta fórmula es que la potencia transmitida va a ir disminuyendo con el cuadrado de la distancia 'r'.

Por tanto, la distancia va a ser un factor muy importante para recibir más o menos potencia.

## 4.2.- Penetración de la señal. Absorción

Cuando la señal transmitida penetra en un objeto, la señal se verá atenuada en mayor o menos medida dependiendo de las características de absorción del objeto, su temperatura y la frecuencia de la señal.

Cada material tiene asociada una constante de atenuación,  $a$  (dB/m). Por ejemplo, la constante de atenuación del agua a temperatura ambiente para una señal de 2.4 GHz es aproximadamente 330 dB/m. El cuerpo humano está compuesto por un 70% de agua, por lo que atenuará las señales significativamente, por lo que saber si una red va a trabajar en un entorno con personas o sin ellas será un factor importante a tener en cuenta, como se verá en el apartado 7 de este trabajo.

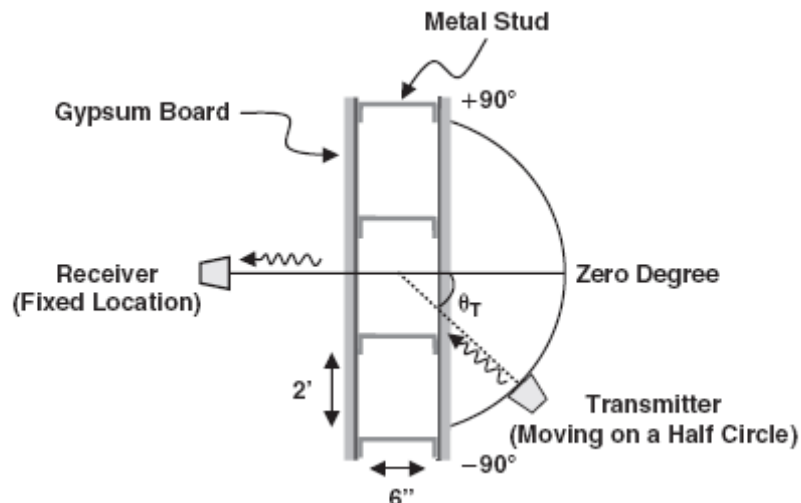
El efecto de la frecuencia de la señal en la penetración depende del tipo de material. Generalmente, considerando los tipos de materiales que se encuentran en una vivienda o en un edificio de oficinas, cuanto mayor es la frecuencia, mayor es la atenuación de la señal. Para la banda ISM de 2.4 GHz, en la tabla 4.1 se muestra una tabla con algunas de las atenuaciones de materiales que se pueden encontrar en un entorno como el que trata este trabajo.

OBJETO	ATENUACIÓN
Pared de escayola	3 dB
Pared de cristal con armazón metálico	6 dB
Pared de ladrillos	4 dB
Ventana de oficina	3 dB
Puerta metálica	6 dB
Puerta metálica en pared de ladrillos	12 dB
Cuerpo humano	3 dB

**Tabla 4.1.** Adquisición de los datos y actuación en un sensor.

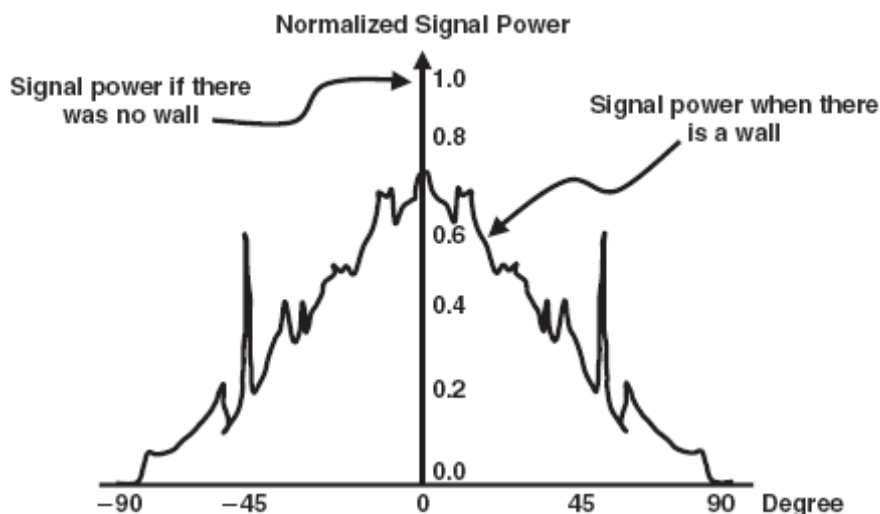


La penetración de la señal en un objeto depende del ángulo al que la señal llega a la superficie del objeto. Considerando una pared de yeso, que se usa habitualmente en las construcciones como pared interior, en la figura 4.1 se puede observar un esquema con un transmisor a un lado de la pared y un receptor al otro lado.



**Figura 4.1.** Medida de la penetración de señal en una pared de yeso. Figura obtenida de [2].

La señal alcanza la pared con un ángulo  $\theta_T$ . Este ángulo va cambiando al ir moviendo el transmisor por un semicírculo. La potencia de la señal recibida se mide al otro lado de la pared. Los datos obtenidos para cada posición del transmisor se pueden representar en una gráfica como la que se muestra en la figura 4.2. La máxima potencia recibida se da cuando la señal alcanza la pared con un ángulo de  $0^\circ$ , y va disminuyendo conforme el ángulo se va acercando a  $-90^\circ$  y a  $+90^\circ$ .

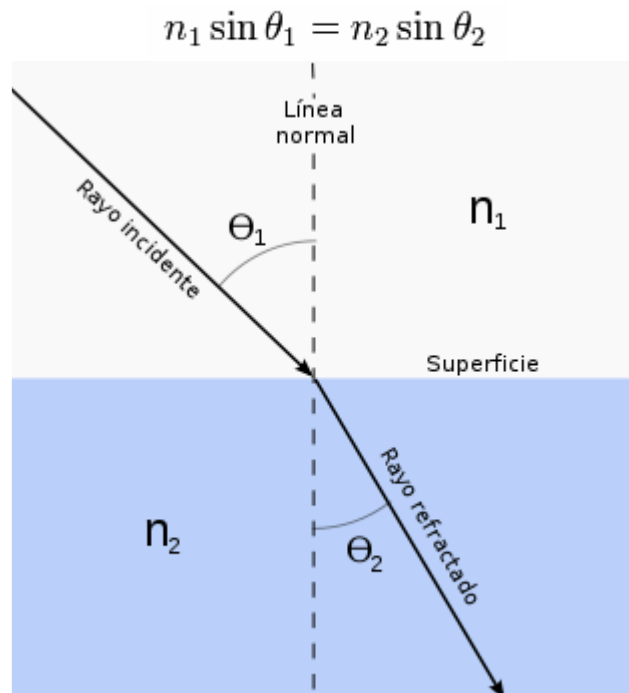


**Figura 4.2.** Potencia de señal recibida para los ángulos de transmisión mostrados en la figura 4.1. Figura obtenida de [2].

Se debe tener en cuenta que estas señales, y las ondas de radio en general, al pasar de un medio a otro, sufren una desviación en su trayectoria. Este fenómeno se llama refracción, y aparece siempre que una onda electromagnética pasa de un medio físico con un índice de refracción a otro con otro índice de refracción distinto.

El índice de refracción es la relación entre la velocidad de propagación de la onda en un medio de referencia, que en el caso de las ondas electromagnéticas es el vacío, y la velocidad de propagación de la misma onda en el medio que corresponda.

La relación entre la onda incidente y la onda refractada nos la da la ley de Snell, cuya fórmula y esquema se muestran en la siguiente figura:

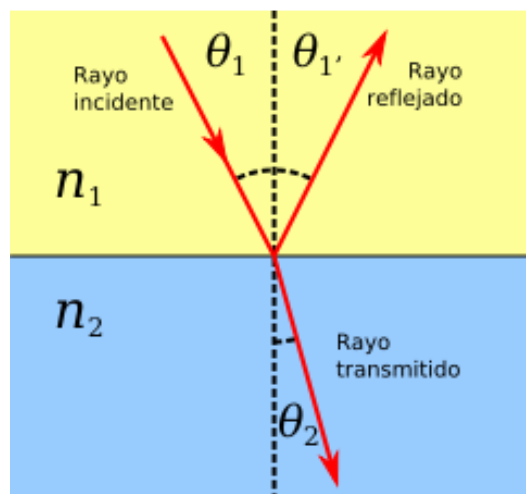


**Figura 4.2.** Ley de Snell, refracción.

### 4.3.- Reflexión, Dispersión y Difracción

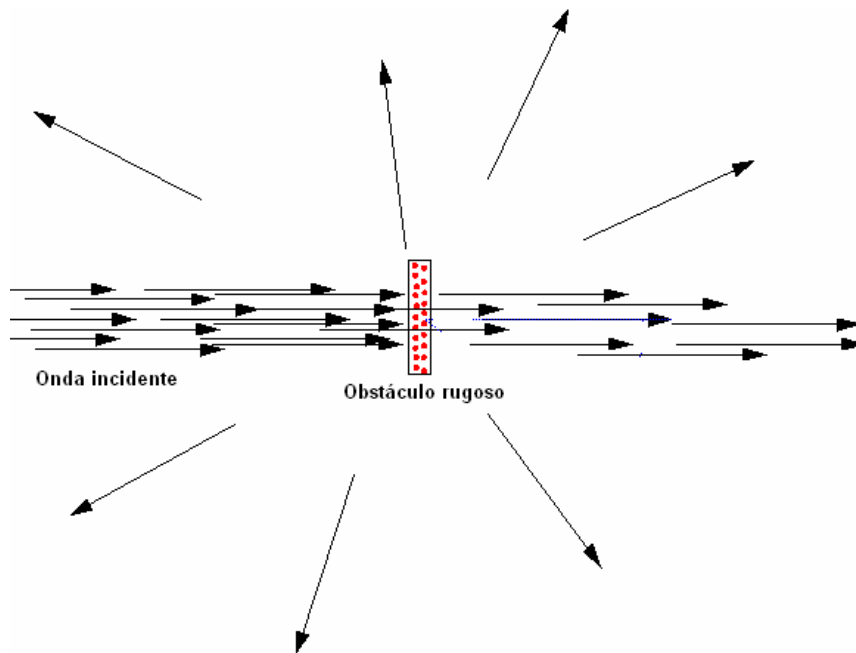
Los fenómenos de la reflexión, dispersión y difracción son muy importantes en las comunicaciones inalámbricas, sobre todo en entornos como los que se van a estudiar en este trabajo, puesto que la presencia masiva de objetos, paredes y suelos van a producir estos efectos en gran cantidad.

Esta presencia de objetos y elementos en nuestros escenarios producen principalmente reflexiones en las superficies en las que inciden las ondas electromagnéticas transmitidas. Si estos elementos fueran conductores perfectos, el 100% de la onda sería reflejada. Pero ni siquiera los elementos metálicos van a ser conductores perfectos, y al igual que en los elementos dieléctricos, se generará una onda transmitida (refractada) y una onda reflejada, como muestra la figura 4.3. A diferencia de la onda transmitida, el ángulo que forma la onda reflejada con la normal y el ángulo que forman la onda incidente y la normal, son iguales.



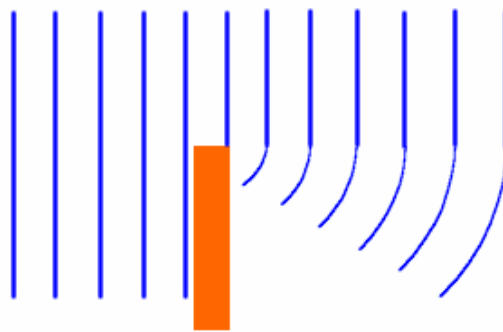
**Figura 4.3.** Ley de Snell, reflexión.

La dispersión, o scattering, ocurre cuando las ondas electromagnéticas inciden sobre una superficie rugosa. La rugosidad de la superficie es diferente para cada onda, ya que depende de la longitud de onda de la señal incidente. Generalmente, si la variación en una superficie es mayor que  $\lambda/8$ , la superficie se considera rugosa y dispersará la señal en varias direcciones (figura 4.4).



**Figura 4.4.** Esquema del fenómeno de dispersión.

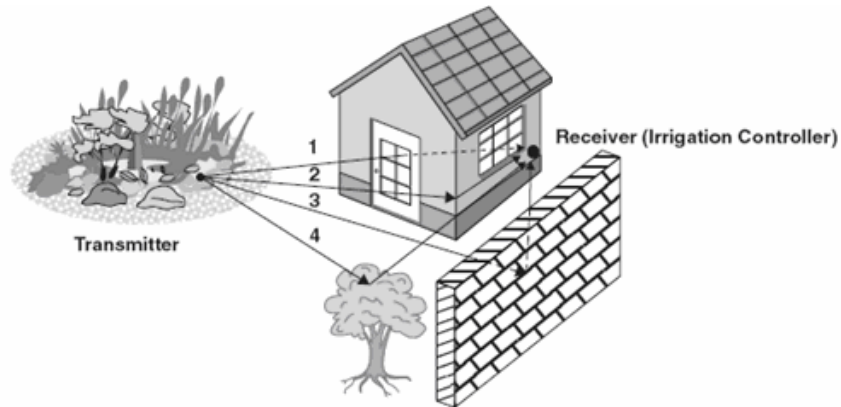
Por último, el fenómeno de la difracción ocurre cuando una onda electromagnética incide en una arista, una esquina, o en un objeto muy fino o estrecho. Al incidir una onda sobre un obstáculo con estas características, el obstáculo se comporta como una nueva fuente para esta onda, rerradiando parte de la energía recibida, posibilitando así la recepción en lugares que antes no era posible por falta de visión directa (figura 4.5).



**Figura 4.5.** Esquema del fenómeno de difracción.

En la figura 4.6 se muestra un ejemplo en el que se dan los 4 fenómenos descritos hasta ahora. El transmisor podría ser un sensor de humedad ubicado en el jardín, que transmite inalámbricamente. En este ejemplo existen 4 diferentes caminos por los que la señal transmitida llega al receptor. El camino 1 representa la penetración de la señal, que pasa a través de las paredes de la casa. El camino 2 representa la difracción, ya que la señal alcanza la esquina que forman dos paredes de la casa y la señal es capaz de cambiar la dirección y llegar al receptor.

El camino 3 representa la reflexión. La señal alcanza una superficie, y una parte de esta señal es reflejada, llegando al receptor. Por último, el camino 4 representa la dispersión, o scattering. Si la frecuencia de la señal es de 2.45 GHz, la longitud de onda es aproximadamente de 12.2 cm, por lo que un obstáculo con variaciones en su superficie mayores de 1.5 cm, que es el caso de un árbol, provoca una dispersión de la señal en varias direcciones. Una de estas direcciones podría ser la del receptor, por lo que la señal también llegaría de este modo.



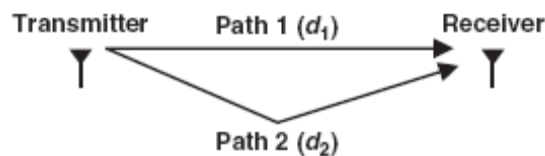
**Figura 4.6.** La señal puede llegar al receptor de varias formas: (1) Penetración, (2) difracción, (3) reflexión y (4) dispersión. Figura obtenida de [2].

## 4.4.- Propagación multitrayecto

A menudo, el entorno en el que se da la comunicación es un entorno complejo, y como se ha visto en el apartado anterior (figura 4.6), la señal transmitida puede llegar al receptor por diferentes caminos. Los escenarios que se van a estudiar en este trabajo son un claro ejemplo de entornos complejos, puesto que existen numerosos elementos que van a interferir en la comunicación, provocando reflexiones, difracciones, dispersión y absorciones de energía. Por lo tanto, los receptores recibirán una señal que se va a componer de la suma de las señales recibidas por diferentes caminos. Cuando la comunicación se da en estas condiciones, se habla de propagación multitrayecto.

Cada camino de propagación diferente hará que la onda recibida llegue con un nivel de potencia y un retraso o adelanto diferentes con respecto a las demás, debido a que las distancias de cada camino serán diferentes. O lo que es lo mismo, cada onda recibida tendrá una fase propia, que al sumarse con las demás, podría crear problemas para una recepción correcta.

Los escenarios estudiados se va a trabajar con situaciones estáticas, es decir, una vez que los transmisores y receptores se coloquen en su lugar, no se van a mover. El caso más simple de propagación multitrayecto será el ilustrado en la figura 4.7.



**Figura 4.7.** Escenario estático multitrayecto. Figura obtenida de [2].

La diferencia de fase (en radianes) entre las dos señales recibidas en el escenario de la figura 4.7 se puede calcular con la siguiente fórmula:

$$\Delta\theta = \frac{2\pi f}{C}(d_1 - d_2) = \frac{2\pi f}{C}\Delta d$$

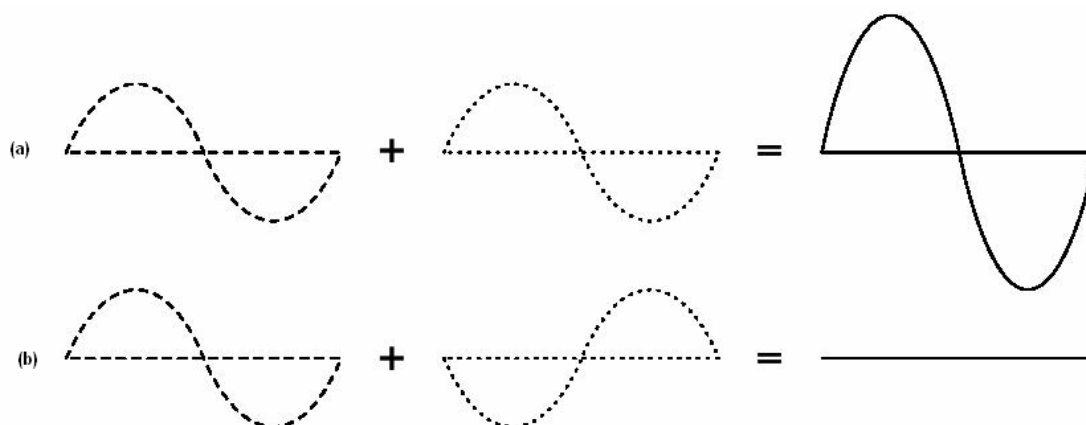
Siendo 'C' la velocidad de la luz en el vacío ( $3 \cdot 10^8$  m/s),  $d_1$  y  $d_2$  las distancias (en metros) de cada camino y 'f' la frecuencia de la señal (en Hz).

En un entorno multitrayecto, varias versiones de la misma señal con diferentes fases y atenuaciones llegarán y se sumarán en el receptor debido principalmente a múltiples reflexiones y difracciones. Esta suma dará lugar a una señal distorsionada, que dependerá de las fases de cada una de las señales recibidas. Esta suma de fases puede tomar un valor entre  $-180^\circ$  y  $+180^\circ$ ,

pudiendo incrementar la potencia de la señal original, pero pudiendo también empobrecer la calidad de esta, llegando a impedir una correcta comunicación. Los dos casos extremos se muestran en la figura 4.8, y son los siguientes:

1- Las señales llegan en fase, es decir, diferencia de fase =  $0^\circ$ , por lo que las señales se sumarán constructivamente.

2- Las señales llegan en contrafase, con una diferencia de fase =  $180^\circ$ , y las señales se sumarán destructivamente.



**Figura 4.8.** Suma de fases, (a) en fase, (b) en contrafase.

Por tanto, siempre existe la posibilidad de que en algunos lugares las señales que llegan se cancelen parcial o totalmente, impidiendo la transmisión de datos.

En escenarios estáticos, como son los estudiados en este trabajo, el impacto de este fenómeno va a ser menor que para escenarios donde el transmisor o el receptor se muevan, ya que en un escenario estático la suma de las fases de llegada dará siempre un valor constante, mientras que en un escenario dinámico será un valor aleatorio. Aunque esto no es del todo cierto, ya que los entornos nunca son totalmente estáticos, y en cuanto un objeto sea cambiado de lugar o haya personas transitando, se podrían generar nuevos trayectos para la señal, cambiando la aportación de esta a la suma de la fase.

#### [4.4.1.-Desvanecimiento multitrayecto](#)

Como ya se ha comentado, la propagación de las señales radioeléctricas a través de un medio depende de las características físicas de este entorno. Estas características físicas suelen ser variables, por lo que la pérdida de la propagación es una variable aleatoria. Aunque la potencia transmitida media se mantenga

constante, la variabilidad de la pérdida de la propagación hace que la potencia recibida sea también variable.

Al valor medio de la potencia recibida se le llama potencia recibida nominal, y un desvanecimiento es una notable reducción de la potencia de señal recibida con relación a su valor nominal. La diferencia entre este nivel nominal y el nivel recibido se llama profundidad de desvanecimiento, y se expresa en dB. Al intervalo de tiempo que transcurre entre la disminución y la recuperación del valor nominal se le llama duración del desvanecimiento. En la figura 4.9 se puede observar, para su mejor entendimiento, un ejemplo de desvanecimiento.

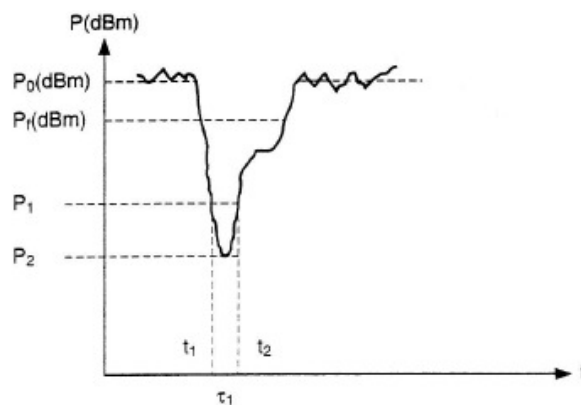


Figura 4.9. Desvanecimiento. Figura obtenida de [5].

$P_0$ (dBm) representa el valor nominal.  $P_f$ (dBm) es el valor de potencia medio cuando se produce un desvanecimiento, que será menor que el valor nominal.  $P_2$  es el nivel mínimo del desvanecimiento, con lo que la profundidad del desvanecimiento será  $P_0$ (dBm) -  $P_2$ . Por último, la duración del desvanecimiento es  $\tau_1$ .

Existen varios tipos de desvanecimiento, pero el que afecta a este trabajo es el desvanecimiento multitrayecto. Estos se generan por la aparición de varios caminos de propagación entre el transmisor y el receptor (reflexión), de forma que se produce una interferencia entre el camino directo y los caminos que alcanzan al receptor con diversos ángulos. La señal resultante de la suma de estas señales tendrá una amplitud variable dependiendo de las amplitudes y fases de las componentes.

El desvanecimiento multitrayecto es muy selectivo en frecuencia, es decir, afecta a un rango de frecuencia muy estrecho. También es muy profundo ( $>20$ dB), aunque de duración muy corta. Debido a esta intensidad y al hecho de ser selectivo en frecuencia, en muchas ocasiones produce una importante atenuación y distorsión en la señal recibida, afectando mucho a la calidad de la comunicación.





El estudio probabilístico de este fenómeno depende de las componentes que llegan al receptor. Cuando existe una componente dominante, la señal resultante suele distribuirse según una estadística de Rice. Por el contrario, si las componentes tienen amplitudes similares, la resultante seguirá una distribución Rayleigh. Esta última situación se produce cuando no existe transmisión directa debido a que el trayecto de propagación directo está obstruido por algún objeto.

En la siguiente figura se muestra un resumen con las características de los desvanecimientos multitrayecto:

Profundidad	Muy profundo (> 20 dB)
Duración	Rápido
Característica espectral	Selectivo
Distribución probabilística	Rayleigh/Rice
Dependencia temporal	Puntual

**Tabla 4.1.** Características de los desvanecimientos multitrayecto. Tabla obtenida de [5].

#### [4.4.2.- Modelos de propagación en interiores](#)

Debido a que durante el estudio habrá escenarios en los que se requiera el uso de modelos de radiopropagación, concretamente el de trazado de rayos, en este apartado se hará una breve descripción sobre este tipo de modelos, aunque existen más.

Un modelo de radiopropagación es un conjunto de expresiones matemáticas, diagramas y algoritmos usados para representar y predecir las características radio de un entorno determinado. Este tipo de modelos se pueden clasificar en empíricos o estadísticos, teóricos o deterministas o una combinación de estos dos (semi-empíricos).

Los modelos empíricos o estadísticos basan sus predicciones en conjuntos de mediciones realizadas previamente. Los modelos teóricos lo hacen en los principios fundamentales de los fenómenos de propagación de ondas de radio. Por último, los modelos deterministas hacen uso de las ecuaciones de Maxwell para el cálculo computacional de las características de la propagación que se quieran estudiar. De este último tipo es el modelo de trazado de rayos.



#### 4.4.2.1-Modelos de propagación deterministas

Estos modelos se basan en los principios fundamentales de la física relacionados con la propagación de ondas de radio. Toda la base teórica que gobierna estos fenómenos electromagnéticos está desarrollada a partir de las ecuaciones de Maxwell, que se pueden representar de la siguiente manera:

$$\begin{aligned}\nabla \times \vec{H} &= \vec{J} + \frac{\partial \vec{D}}{\partial t} \\ \nabla \times \vec{E} &= -\frac{\partial \vec{B}}{\partial t} \\ \nabla \cdot \vec{D} &= \rho \\ \nabla \cdot \vec{B} &= 0\end{aligned}$$

**Figura 4.10.** Ecuaciones de Maxwell.

Usando estas cuatro ecuaciones se pueden resolver todos los problemas electromagnéticos, obteniendo una solución concreta para cada punto del espacio. Las técnicas deterministas consiguen sus resultados resolviendo alguna aproximación a estas ecuaciones. Cabe destacar que a pesar de que parezcan ecuaciones sencillas, las expresiones que se derivan de ellas son muy complejas, y requieren gran carga computacional para poder resolverlas, ya que además deben ser aplicadas en todos los puntos del espacio, teniendo en cuenta todas las condiciones de contorno para cada punto.

#### 4.4.2.2-Modelos basados en técnicas de trazado de rayos (Ray Tracing)

Existen numerosas técnicas de modelado deterministas. Una de las más utilizadas en entornos interiores es la basada en Óptica Geométrica (*Geometrical Optics*). Las técnicas basadas en trazado de rayos se fundamentan en la aplicación de esta Óptica Geométrica.

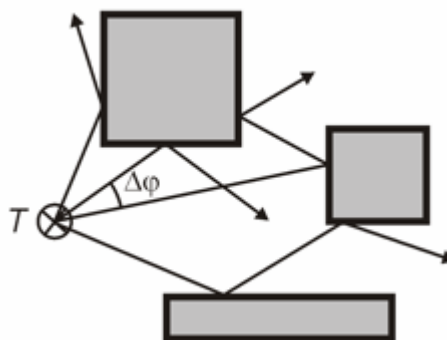
Este método es uno de los más antiguos que se utilizan para alta frecuencia, ya que se desarrolló originalmente para analizar la propagación de la luz. En esta teoría se considera el transporte de energía en rayos, sin precisar si la energía se propaga en forma de ondas o de partículas. La teoría de la Óptica Geométrica se basa en los siguientes conceptos:

- Las líneas de flujo de energía o rayos son rectas en un medio homogéneo. En medios heterogéneos, siguen el camino de longitud mínima (principio de Fermat).
- Los rayos son reflejados por la superficie conductora según la ley de Snell. Por tanto, a un punto determinado llegarán únicamente los rayos provenientes de unos puntos de reflexión concretos, y no del resto de la superficie de los objetos.
- La variación de la amplitud del campo a lo largo del recorrido de los rayos está determinada por la ley de conservación de la energía.

En estos algoritmos, la señal se propaga al igual que si fuesen rayos ópticos (aproximación del frente de onda), lo que hace que el análisis sea más sencillo. Los fenómenos que se contemplan en la propagación son principalmente la reflexión, refracción y difracción. Aunque también existen otros fenómenos como el *Scattering* (Dispersión) que algunos algoritmos también contemplan.

Uno de los principales modos de trazar los rayos es el método de lanzado de rayos. Este es el modo que se usa en el programa con el que se va a trabajar en este estudio. Esta técnica, también conocida como *Ray Launching*, consiste en lanzar un número finito de rayos desde el transmisor. La distribución de los rayos en el ángulo sólido deseado debe ser uniforme, cubriéndolo en su totalidad.

Para cada rayo se consideran múltiples reflexiones y refracciones hasta alcanzar el receptor. El número de estas reflexiones y refracciones dependerá de la naturaleza de los obstáculos que encuentre en el camino y del nivel de la señal transmitida. La figura 4.12 se muestra un esquema de esta técnica.



**Figura 4.11.** Método de *Ray Launching*.

Las soluciones que se obtienen con los modelos basados en Óptica Geométrica son relativamente simples, aunque la calidad de estos resultados depende del tamaño eléctrico de los objetos. Si son grandes, los resultados serán mejores.



# 5.-Redes de sensores inalámbricos.

## ZigBee

### 5.1.- Redes de sensores

#### 5.1.1.-Introducción

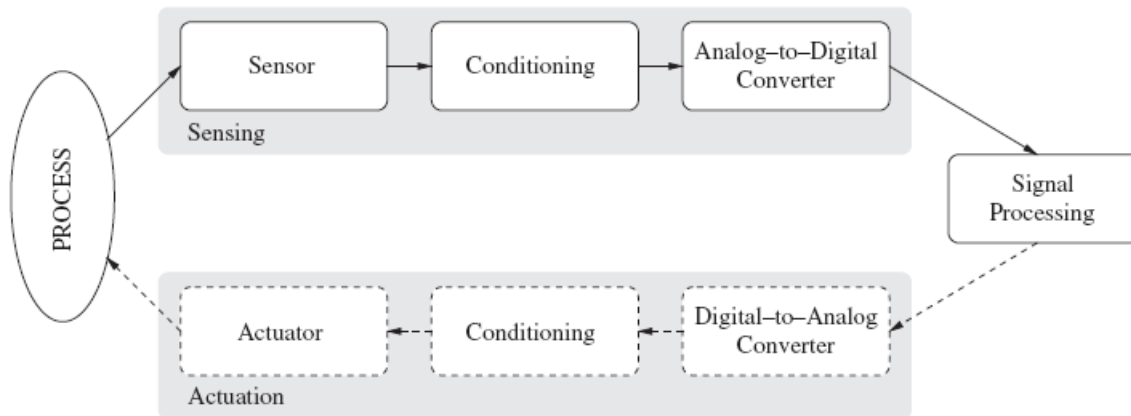
Los sensores son dispositivos que relacionan el mundo físico (real) con el mundo digital capturando y revelando un fenómeno físico y convirtiéndolo a un formato (señal eléctrica) que pueda ser procesado y/o almacenado. Integrados en distintos dispositivos y entornos, nos proporcionan un gran beneficio: Pueden ayudarnos a evitar fallos catastróficos en las infraestructuras, conservar recursos naturales, aumentar la productividad, aumentar la seguridad y nos dan la posibilidad de crear infinidad de nuevas aplicaciones.

Los grandes avances en tecnologías como los sistemas microelectromecánicos, VLSI (Very Large Scale Integration) y las comunicaciones inalámbricas, contribuyen a extender el uso de sistemas distribuidos de sensores. Por ejemplo, el gran desarrollo en tecnologías de semiconductor incrementa la capacidad de procesado y disminuye el tamaño. Esta miniaturización en las tecnologías de computación y de sensores permite el desarrollo de sensores, actuadores y controladores de tamaño muy pequeño, de bajo consumo y de bajo coste.

Además, continuamente se están buscando y desarrollando nuevas aplicaciones en cada vez mayor número de áreas. Aunque las aplicaciones militares y sistemas aeroespaciales dominan el mercado, otras áreas como la monitorización y protección de infraestructuras civiles (túneles, puentes,...), las redes de distribución eléctricas o las infraestructuras de tuberías están aumentando día a día. De hecho, ya existen redes de cientos de sensores para monitorizar los niveles de polución en grandes áreas geográficas, para recoger información sobre el estado estructural de puentes, y para controlar los niveles de pesticidas, humedad y fertilizantes en cultivos.

En la figura 5.1 se muestra un ejemplo de los pasos que se dan a la hora de sensar una magnitud física. Dicho fenómeno (Process) físico es captado por el sensor. La señal eléctrica resultante normalmente no está preparada para procesarla directamente, por lo que se debe acondicionar (filtrado, amplificación,

etc.) para su posterior análisis. Por ejemplo, las señales suelen necesitar amplificación (o atenuación) para ajustar la magnitud de la señal al rango de entrada necesario del convertidor analógico-digital. Una vez digitalizada la señal, ya está lista para procesarla, almacenarla o visualizarla.



**Figura 5.1.** Adquisición de los datos y actuación en un sensor. Figura obtenida de [3].

Muchas redes de sensores inalámbricos también incluyen actuadores con los que se controla directamente la magnitud física medida. Dependiendo de los datos que se hayan obtenido, se mandará una señal al actuador, previamente acondicionada adecuadamente, convirtiéndola en señal de entrada de dicho actuador (Figura 5.1). A continuación, el actuador realizará la tarea oportuna, cambiando el valor de la magnitud física correspondiente, formando un lazo cerrado de control. Un actuador puede ser una válvula que controla el flujo de agua caliente, un motor que cierra o abre una puerta o ventana, o una bomba que controla la cantidad de combustible que se inyecta en una máquina.

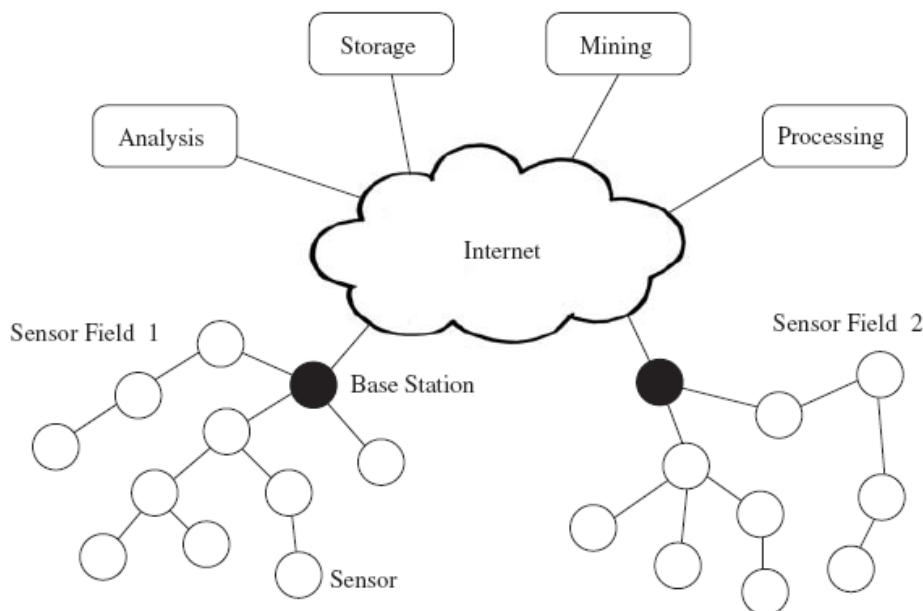
### 5.1.2.-Redes de sensores inalámbricos

Aunque actualmente la gran mayoría de sensores están conectados directamente mediante cableado a los controladores y estaciones de procesado, cada vez son más los sensores que se comunican inalámbricamente con una estación centralizada donde se realiza el procesado de los datos obtenidos. Esto es muy importante, ya que muchas aplicaciones requieren cientos o incluso miles de sensores, que además se tienen que colocar en lugares remotos y difícilmente accesibles.

Por tanto, un sensor inalámbrico no es solamente un componente para sensar, sino que tiene que ser capaz de procesar señales, de comunicarse y de realizar otras funciones específicas según la aplicación requerida en cada caso.

Muchas veces, el sensor no sólo es responsable de los datos que ha adquirido, sino que participa en la creación de la red, la búsqueda de caminos y la transmisión de los datos de otros sensores.

Cuando muchos sensores monitorizan conjuntamente un mismo entorno físico, se dice que forman una red de sensores inalámbricos, o WSN (Wireless Sensor Network). Los sensores se comunican entre sí y con la estación base (BS) mediante comunicación radio, permitiendo difundir los datos que han obtenido hasta llegar a los sistemas de procesado, visualización, análisis y almacenamiento, que generalmente estarán en lugares alejados de la red. En la figura 5.2 se muestran dos zonas monitorizadas, que se encuentran geográficamente en regiones diferentes. Ambas zonas están conectadas a Internet mediante sus estaciones base (BS).



**Figura 5.2.** Redes de sensores inalámbricos. Figura obtenida de [3].

La capacidad de los sensores en una WSN puede variar mucho, desde un nodo simple que monitorice un único fenómeno físico, hasta dispositivos complejos que combinen diferentes técnicas de detección (acústica, óptica, magnética,...). También pueden diferir en la tecnología que usan para transmitir los datos (ultrasonidos, infrarrojos o radiofrecuencia), con lo que también variará la velocidad de transmisión. Lo mismo ocurre con la capacidad de procesado, enrutamiento y almacenamiento. Finalmente, también hay dispositivos que soportan tecnologías adicionales, como por ejemplo GPS (Global Positioning System). Sin embargo, tales sistemas consumen demasiada energía para usarlos como nodos-sensores de bajo consumo y bajo coste.



### 5.1.2.1-Estándares para una WSN

La conocida familia de estándares IEEE 802.11 que fue introducida en el año 1997, es la tecnología de redes inalámbricas más común en la actualidad. IEEE 802.11b y IEEE 802.11g trabajan en la banda de 2.4 GHz y IEEE 802.11a en la de 5 GHz. IEEE 802.11 fue usada en las primeras redes de sensores inalámbricos y aún hoy en día puede encontrarse en aplicaciones en las que el ancho de banda requerido es muy alto. Pero debido al elevado consumo de energía de estas redes, hace que el estándar IEEE 802.11 no sea idóneo para las redes de sensores de bajo consumo. Además, la velocidad de transmisión que ofrece es típicamente mucho mayor de la necesaria. Esto hizo necesario el desarrollo de protocolos que satisficieran las necesidades de las redes de bajo consumo y baja tasa de transmisión. Por ejemplo, el protocolo IEEE 802.15.4, que es el protocolo con el que he trabajado en este proyecto y que se explicará con más detenimiento más adelante, fue creado específicamente para comunicaciones de corto alcance en redes de sensores de bajo consumo.

### 5.1.2.2.-Características clave y restricciones en el diseño de WSNs

Aunque los distintos tipos de redes inalámbricas comparten muchas características, las redes de sensores inalámbricas requieren algunas características muy concretas, por lo que es necesario un diseño específico. A continuación comentaré las más importantes para una WSN.

**-Energía:** Los nodos-sensores deben consumir muy poca energía. Típicamente son alimentados por baterías, que deben ser recargadas o cambiadas cuando se acaban. Para algunos sensores, ninguna de las dos opciones es válida, sino que cuando se gasta la batería, hay que cambiarlos por otros nuevos.

Dependiendo de si la batería puede ser recargada o no, la estrategia a seguir cambiará significativamente. Para las baterías no recargables hay dos opciones. La primera son los que disponen de una batería que al acabarse se puede cambiar por otra. La segunda opción son los sensores que al acabarse la batería, tienen que ser reemplazados por otros. El tiempo de vida de estos dispositivos se denomina mission time. El mission time dependerá de la aplicación: para monitorizar los movimientos de un glaciar, probablemente se necesitarán sensores con varios años de vida. En cambio, para un escenario militar, con pocas horas o días será suficiente.

En consecuencia, la primera y más importante característica en el diseño de sensores para una WSN es la eficiencia energética.

**-Autogestión (Self-Management):** En muchos casos, la WSN tendrá que operar en áreas remotas, y muchas de ellas con difícil accesibilidad, donde no



habrá infraestructuras ni la posibilidad de mantenimiento o reparación. Por tanto, es muy importante que la red sea capaz de autoconfigurarse, que los nodos puedan colaborar entre sí y que puedan adaptarse a fallos y a cambios en el entorno sin la necesidad de intervención humana. Podemos distinguir dos tipos de modos de operación dentro de los autogestionables:

*-Despliegue Ad Hoc:* Muchas aplicaciones de WSNs no necesitan que los nodos-sensores estén en una localización determinada. Esta característica es muy importante en los casos en que la zona en que hay que desplegar la red es inaccesible, como por ejemplo en la evaluación de desastres, donde los sensores se sueltan desde aviones y varios de estos sensores podrían nunca llegar a funcionar debido al golpe o a que han caído muy lejos. Pero si estos nodos son capaces de autoconfigurarse, determinar su propia posición, comunicarse con los nodos más cercanos y configurar la red, no habrá problema.

*-Unattended Operation:* En este caso, los nodos sí que necesitan estar en una situación concreta, y tienen que ser desplegados según esa configuración. Una vez desplegados, deben operar sin intervención humana, adaptándose a los cambios en el entorno. Por ejemplo, siendo capaz de regular la potencia de transmisión para que el receptor mantenga el nivel de potencia recibida cuando cambia la meteorología.

**-Wireless Networking:** A la hora de configurar la red, existen varios parámetros que hay que tener en cuenta. Por ejemplo, la atenuación limita el rango de la señal radio. Al propagarse por un medio y al traspasar obstáculos, el nivel de potencia se reduce. La relación entre la potencia recibida y la potencia transmitida de una señal RF se puede expresar usando la siguiente fórmula:

$$P_r \propto \frac{P_t}{d^2}$$

Donde se observa que la potencia recibida ( $P_r$ ) es proporcional a la inversa del cuadrado de la distancia ( $d$ ) desde la fuente de la señal. En consecuencia, un incremento de la distancia entre un nodo-sensor y una estación base (BS) incrementa rápidamente la necesidad de aumentar el nivel de potencia transmitido (para mantener el mismo nivel de potencia recibida). Por tanto, desde el punto de vista de la eficiencia energética, es mejor dividir una distancia grande en varios saltos de menor longitud, siempre y cuando la comunicación multi-hop sea posible.

Pero en la comunicación multi-hop los nodos de la red deben cooperar entre sí para identificar las rutas. Es decir, necesitan cierta inteligencia para realizar más tareas, lo que implica un mayor consumo de energía. Para





compensar este consumo, se usan estrategias para reducirlo, manteniendo la transmisión y recepción radio apagada cuando los nodos-sensores no tienen información propia que transmitir. En este tipo de sistemas, hay que tener presente que dependiendo de la aplicación y la configuración de la red, habrá nodos que no puedan 'apagarse', ya que formarán parte del backbone de la red y el flujo de datos de la red pasará por ellos.

**-Administración descentralizada:** Las WSNs a gran escala y las limitaciones de energía hacen inviable el uso de algoritmos centralizados para implementar soluciones para administrar la red, como la administración de la topología o el enrutamiento.

En su lugar, los nodos-sensores colaboran con sus vecinos para tomar decisiones locales, sin necesidad de una regla global. Estos algoritmos descentralizados no suelen ser óptimos, pero sí más eficientes respecto al consumo energético que los algoritmos centralizados.

Si tomamos como ejemplo el enrutamiento, una estación base puede recibir la información de todos los sensores, establecer rutas óptimas e informar a cada nodo su ruta. Si la topología cambia frecuentemente, la sobrecarga puede ser significativa. Si en este caso los nodos fueran capaces de tomar decisiones sobre el enrutamiento con información limitada a su entorno más cercano, se reduciría la sobrecarga significativamente, aunque no fueran las rutas más óptimas desde el punto de vista global de la red.

**-Limitaciones en el diseño:** El objetivo primario en el diseño de sensores inalámbricos es crear dispositivos cada vez más pequeños, más baratos y más eficientes. Debido a estas necesidades, muchas veces se hace inviable la integración de otros componentes deseables, como el GPS.

Estas limitaciones y requerimientos también tienen su impacto en el diseño del software y del hardware. Por ejemplo, las tablas de rutas que contienen las entradas para cada uno de los destinos potenciales en la red suelen ser demasiado grandes y requerirían que cada sensor tuviera una gran memoria. En vez de eso, con una memoria mucho menor, pueden guardar una pequeña cantidad de información (sólo de sus vecinos), ahorrando espacio y energía.

En general, muchas soluciones de software (sistema operativo, middleware, protocolos de red) tienen que ser diseñados para operar eficientemente sobre hardware limitado en recursos.

**-Seguridad:** Muchas redes de sensores inalámbricos transmiten información sensible, y cuando operan remotamente y sin intervención humana, se exponen fácilmente a intrusiones y ataques. Además, el uso de comunicaciones inalámbricas hace más fácil aún un posible ataque.



Aunque existen numerosas técnicas y soluciones para prevenir estos ataques, la mayoría de ellas necesitan un significativo coste computacional y requerimientos de memoria, que muchas veces no pueden ser satisfechos debido a la limitación de recursos del que disponen los sensores.

Por tanto, las WSNs requieren de nuevas soluciones para el establecimiento de claves, autenticación de los nodos, etc.

**-Otros retos:** Existen una gran cantidad de retos, que aparecen dependiendo de la aplicación, que afectarán a la hora de diseñar los nodos-sensores y las WSN. Por ejemplo, habrá ocasiones en las que los sensores deberán ir montados en objetos en movimiento, como vehículos o robots, donde constantemente el entorno cambiará rápidamente.

También habrá casos en los que los sensores tendrán una función muy específica y que necesiten gran calidad en sus prestaciones, como por ejemplo muy bajas latencias o un gran throughput (video-sensores).

Por último, cabe destacar que a diferencia que las redes tradicionales, en las WSN no suelen estar basadas en estándares establecidos, sino que muchos protocolos y mecanismos son soluciones propietarias.

Como ya sabemos, la estandarización es muy importante para la interoperabilidad de dispositivos y para facilitar el diseño y desarrollo de las aplicaciones de WSN. Por lo que un reto clave reside en la estandarización y armonización entre los estándares que compiten entre sí.

A modo de resumen, en la tabla 5.1 se muestran algunas de las diferencias más características entre las redes tradicionales y las WSN.

Traditional networks	Wireless sensor networks
General-purpose design; serving many applications	Single-purpose design; serving one specific application
Typical primary design concerns are network performance and latencies; energy is not a primary concern	Energy is the main constraint in the design of all node and network components
Networks are designed and engineered according to plans	Deployment, network structure, and resource use are often ad hoc (without planning)
Devices and networks operate in controlled and mild environments	Sensor networks often operate in environments with harsh conditions
Maintenance and repair are common and networks are typically easy to access	Physical access to sensor nodes is often difficult or even impossible
Component failure is addressed through maintenance and repair	Component failure is expected and addressed in the design of the network
Obtaining global network knowledge is typically feasible and centralized management is possible	Most decisions are made localized without the support of a central manager

**Tabla 5.1.** Comparación de las redes tradicionales y las WSN. Figura obtenida de [3].

## 5.2.- ZigBee

ZigBee es el nombre del estándar más utilizado para la implementación de redes inalámbricas de bajo coste, baja tasa de transmisión, corto alcance y bajo consumo de energía. Además, es el estándar que he usado para realizar el proyecto fin de carrera, por lo que me centraré exclusivamente en él.

### 5.2.1.-Introducción

Los dispositivos basados en ZigBee operan en las bandas de 868 MHz, 915 MHz y 2.4 GHz, y su velocidad de transmisión máxima es de 250 Kbps. Como ya he comentado anteriormente, ZigBee está diseñado para aplicaciones de baja tasa de datos, bajo coste y bajo consumo energético, ya que generalmente los dispositivos se alimentan mediante baterías que será complicado recargar o cambiar. De hecho, en muchas aplicaciones el tiempo total que el dispositivo inalámbrico está activo es muy limitado. El dispositivo pasa la mayor parte del tiempo en un modo de ahorro de energía llamado sleep mode, pudiendo alargar la vida de la batería hasta varios años.

El estándar ZigBee lo creó y lo desarrolla The ZigBee Alliance, grupo abierto a todo el mundo creado en 2002, el cual está formado por cientos de compañías, muchas de ellas líderes de mercado en sus respectivas áreas (ver Figura 5.3).



**Figura 5.3.** Miembros de The ZigBee Alliance. Figura obtenida de [1].

El estándar ZigBee adoptó del estándar IEEE 802.15.4 los protocolos para su capa física y capa MAC (Medium Access Control). Por lo que un dispositivo que cumple la norma ZigBee será compatible con el estándar IEEE 802.15.4 (ver Figura 5.4).

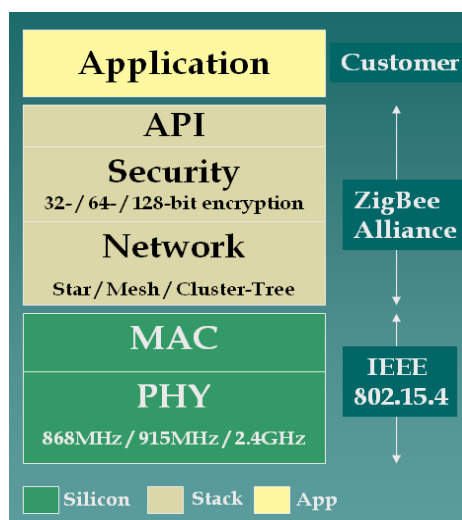


Figura 5.4. ZigBee y el estándar IEEE 802.15.4.

## 5.2.2.-Características de ZigBee

En este apartado analizaremos una por una las características más representativas del estándar ZigBee, que son las siguientes:

- Alta fiabilidad.
- Bajo coste.
- Bajo consumo de energía.
- Altamente seguro.
- Estándar abierto.
- Baja tasa de transmisión de datos.

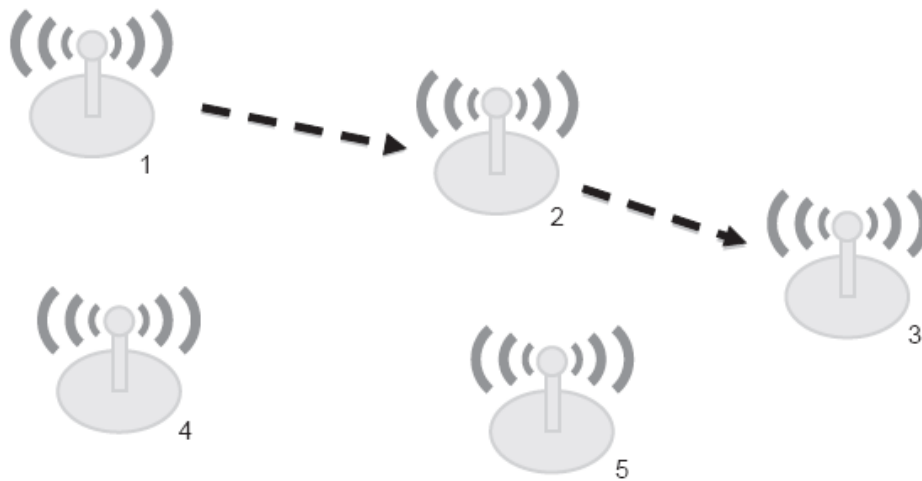
### 5.2.2.1.-Alta fiabilidad (Highly reliable)

Las comunicaciones inalámbricas son de por sí muy vulnerables. Pueden ser bloqueadas por algún metal o agua. Varían mucho debido a factores como el diseño de la antena, la potencia de transmisión, las condiciones meteorológicas y las interferencias.

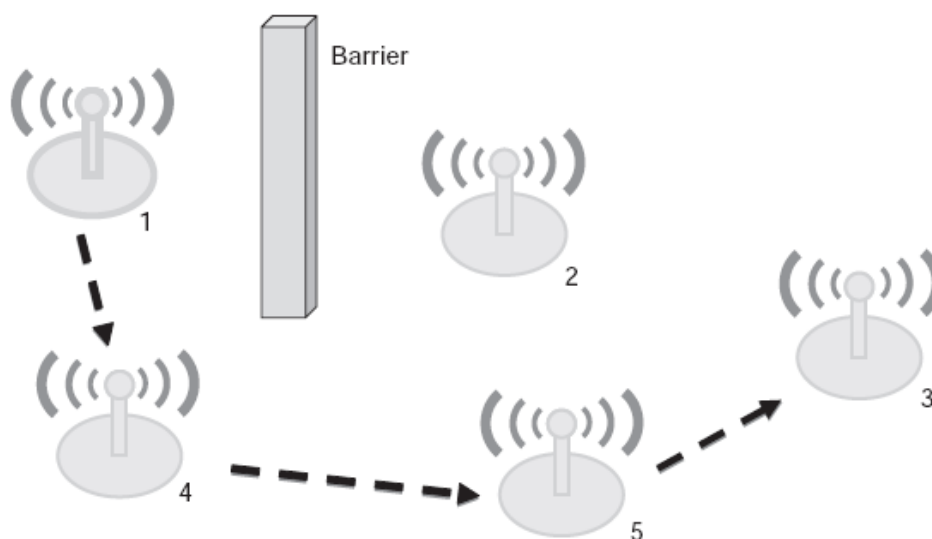
ZigBee proporciona alta fiabilidad de diferentes maneras:

- IEEE 802.15.4 con O-QPSK (Offset-Quadrature Phase-Shift Keying) y DSSS (Direct Sequence Spread Spectrum) ofrece un excelente rendimiento en entornos con SNR baja.
- CSMA-CA (Carrier Sense Multiple Access Collision Avoidance), con lo que se evitan colisiones en el canal, ya que ZigBee antes de transmitir escucha, y sólo transmitirá cuando el canal esté libre.

- CRC de 16 bits, usado en cada paquete que se transmite, así asegurando que cada bit de los datos transmitidos es correcto.
- ACKs en cada salto. Cada paquete puede ser retransmitido hasta 3 veces (un total de 4 transmisiones).
- Topología de red tipo Mesh para buscar la ruta más fiable. Cuando una ruta ya establecida se corta, ZigBee es capaz de buscar una ruta alternativa, como se muestra esquemáticamente en las figuras 5.5 y 5.6 (El nodo 1 transmite al nodo 3). Adicionalmente, ZigBee proporciona broadcasting y multicasting.
- ACKs End-to-end para verificar la recepción de los datos por parte del destinatario final. ZigBee desecha cualquier paquete duplicado, por lo que la aplicación no deberá hacerlo.



**Figura 5.5.** Mesh Networking con ruta establecida. Figura obtenida de [1].



**Figura 5.6.** Mesh Networking con ruta alternativa. Figura obtenida de [1].

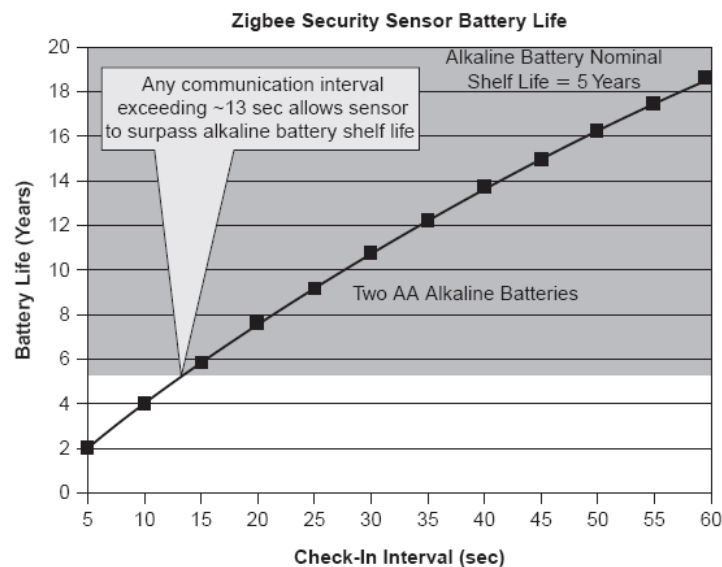
### 5.2.2.2.-Bajo coste

Como se vio en el apartado 3, el coste de los chips o transceivers puede ser muy bajo, menos de 1 euro. Los módulos ZigBee, oscilan entre los 15 y 25 euros (figura 3.1).

Pero también hay que tener en cuenta que la banda de frecuencias de 2.4 GHz, que es la banda más común en la que trabaja ZigBee, es libre a nivel mundial, por lo que no se debe pagar licencia.

### 5.2.2.3.-Bajo consumo de energía

Un dispositivo ZigBee puede funcionar varios años con un par de baterías AA. La principal razón para que ZigBee consuma tan poco, es su duty cycle. Un nodo en una red no tiene la necesidad de estar permanentemente conectado a la red para permanecer en ella. Por ejemplo, un sensor de temperatura puede que tenga que transmitir sólo una vez por hora, a no ser que haya un cambio brusco en la temperatura. La siguiente gráfica muestra aproximadamente la vida de la batería frente al intervalo de tiempo entre comunicaciones:



**Figura 5.7.** Tiempo de vida de la batería (AA) en ZigBee. Figura obtenida de [1].



#### 5.2.2.4.-Altamente seguro

Para dotar a la red de la máxima seguridad posible, ZigBee usa el estándar AES-128 del NIST (National Institute of Standards and Technology). AES son las siglas de Advanced Encryption Standard. Este estándar es un bloque de cifrado que encripta los paquetes de forma que es muy difícil que alguien ajeno a la comunicación pueda descifrar su contenido, además de ofrecer autenticación para asegurar que los paquetes son propios de la red. Las razones por la que se ha adoptado para ZigBee son varias, pero las más significativas son que es un estándar fiable y reconocido internacionalmente, está libre de incumplimientos de patentes y que se puede implementar en procesadores de 8 bits.

#### 5.2.2.5.-Estándar abierto

ZigBee es un estándar global abierto. Se puede descargar gratis de la página web [www.ieee.org](http://www.ieee.org). La anteriormente mencionada The ZigBee Alliance, además de desarrollar el estándar, también especifica varios test para que las compañías puedan desarrollar sus dispositivos respetando y cumpliendo el estándar. Así se asegura la compatibilidad entre dispositivos desarrollados por distintas compañías.

Que los distintos dispositivos y aplicaciones funcionen perfectamente entre sí es indispensable, ya que una red ZigBee podría estar formada por productos de diferentes vendedores.

#### 5.2.2.6.-Baja tasa de transmisión de datos

Hoy en día, en la mayoría de los casos en los que existe una comunicación, se busca desarrollar la tecnología para aumentar la tasa de transmisión de datos. En el caso de ZigBee esto no se cumple. Aunque haya aplicaciones concretas en las que sea necesaria una tasa de transmisión alta, no es lo común, ya que ZigBee está pensado y diseñado para la monitorización y control inalámbrico de algún parámetro.

Como ya sabemos, ZigBee se basa en 802.15.4, donde la máxima velocidad cuando trabajamos a 2.4 GHz es de 250 Kbps. Esta es la máxima velocidad a la que trabaja ZigBee. Más adelante se comentarán más aspectos sobre canales frecuenciales, velocidades de transmisión, etc.



### 5.2.3.-ZigBee y 802.15.4

Como ya se ha comentado en este trabajo, el estándar 802.15.4 fue creado y es actualizado y revisado por el IEEE. Al igual que todos los IEEE 802.x.x, el 802.15.4 define un estándar para redes (802.3:Ethernet, 802.11:WiFi, 802.15.1:Bluetooth). Concretamente, define la capa física (PHY) y la capa de acceso al medio (MAC) de una red de área personal (PAN), inalámbrica y de bajo consumo. Entre otros, estas son las tareas o mecanismos más importantes que define este estándar:

- Mecanismo para el descubrimiento de redes.
- Mecanismo para formar y unirse a redes.
- Mecanismo para cambiar de canal.
- Mecanismo para detectar interferencias y ruido en el canal.
- Método de envío de paquetes con ACK usando CSMA-CA para evitar colisiones (únicamente single-hop).
- Método de envío broadcast, sin ACK y sólo en single-hop.

Lo que 802.15.4 no especifica son la comunicación multi-hop, la asignación de direcciones o la interoperabilidad a nivel de aplicación. Por tanto, si la red que se quiera construir sólo necesita comunicaciones de un salto (single-hop), con dispositivos que cumplan este estándar será suficiente. De hecho, hay compañías que ofrecen aplicaciones de este tipo, que no requieren otro protocolo, como ZigBee.

A parte de ZigBee, existen otros protocolos de red que tienen como base el estándar 802.15.4. A continuación se nombran los más significativos:

-*SimpliciTI*. Este protocolo, desarrollado por Texas Instruments, es muy simple, diseñado para comunicaciones punto a punto, y es extremadamente pequeño (4 Kbyte de flash).

-*Synkro*. Originalmente desarrollado por Freescale Semiconductor, hoy en día es un protocolo emergente, cuyo uso es específico en la electrónica de consumo. Se usa para el control de televisiones, reproductores DVD, etc. Synkro también es punto a punto, pero ofrece un sistema que evita los problemas de interferencias frecuenciales.

-*MiWi*, de MichoChip, es un protocolo peer-to-peer que puede trabajar en topología de malla (mesh) que se diseñó específicamente para trabajar con el dispositivo radio MRF24J40.

-*PopNet*, desarrollado por San Juan Software, es de las soluciones mencionadas hasta ahora es la que más se acerca a las prestaciones que ofrece ZigBee. Soporta broadcasting, unicasting y enrutamiento en topología en malla.

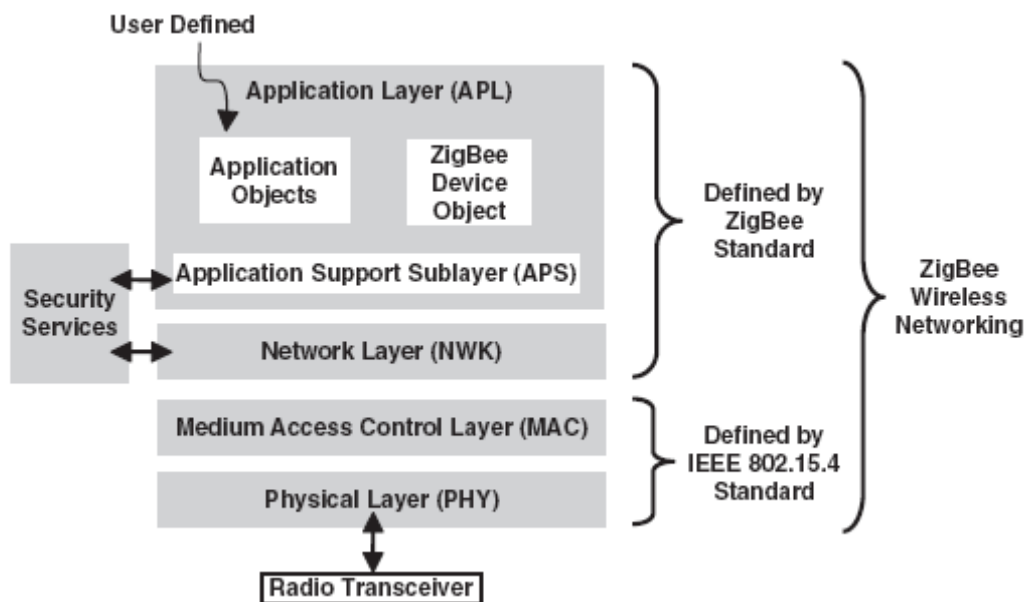


-*TinyOS* es una tecnología para redes de topología en malla escrita en un lenguaje llamado NesC. Este lenguaje parecido a C permite que los sensores sean tratados como objetos. TinyOS fue adoptado por uno de los vendedores de ZigBee, MeshNetics, como fundación de su solución ZigBee.

-*6LoWPAN* fue estandarizado por IETF.org. Es un protocolo de red para 802.15.4 que se parece a IPv6. Compañías como ArchRock (recientemente adquirida por Cisco) estaban promoviendo este estándar para usarlo como interface entre los sensores y WWW.

Todos estos protocolos tienen sus ventajas y sus puntos flacos, pero ZigBee es el principal protocolo basado en 802.15.4, por lo que hoy en día es el más adecuado si queremos aplicaciones con alta interoperabilidad entre dispositivos.

En la figura 5.8 se puede ver la relación entre 802.15.4 y ZigBee. Como ya se ha comentado, IEEE 802.15.4 define las especificaciones de la capa física (PHY) y la capa de acceso al medio (MAC), pero no especifica nada sobre las capas superiores. Por el contrario, el estándar ZigBee define las capas de red, seguridad y de aplicación, adoptando las capas MAC y física definidas por 802.15.4 para crear el protocolo de red de ZigBee.



**Figura 5.8.** Capas del protocolo de red de ZigBee. Figura obtenida de [2].

Cabe destacar que ZigBee no usa todas las especificaciones de las capas MAC y PHY de 802.15.4. Esto permite a las compañías mayor flexibilidad para ofrecer diferentes tipos de soluciones, como por ejemplo soluciones de menor RAM y flash.



### 5.2.4.-Frecuencias y velocidades de transmisión

Hay 3 bandas de frecuencias en la versión más actual de IEEE 802.15.4, del año 2006:

- Banda de 868 MHz: 868-868.6 MHz.
- Banda de 915 MHz: 902-928 MHz.
- Banda de 2.4 GHz: 2400-2483.5 MHz.

La banda de 868 MHz se usa en Europa en varios tipos de aplicaciones, incluyendo las redes inalámbricas de corto alcance. Las otras dos bandas son parte de las bandas frecuenciales de ISM, por lo que no requieren licencia. La banda de 915 MHz es usada principalmente en USA, mientras que la de 2.4 GHz se usa en todo el mundo. La tabla 5.2 muestra más detalles sobre las bandas frecuenciales que usa el estándar 802.15.4.

	Frequency (MHz)	Number of Channels	Modulation	Chip Rate (Kchip/s)	Bit Rate (Kb/s)	Symbol Rate (Ksymbol/s)	Spreading Method
	868-868.6	1	BPSK	300	20	20	Binary DSSS
	902-928	10	BPSK	600	40	40	Binary DSSS
Optional	868-868.6	1	ASK	400	250	12.5	20-bit PSSS
	902-928	10	ASK	1600	250	50	5-bit PSSS
Optional	868-868.6	1	O-QPSK	400	100	25	16-array orthogonal
	902-928	10	O-QPSK	1000	250	62.5	16-array orthogonal
	2400-2483.5	16	O-QPSK	2000	250	62.5	16-array orthogonal

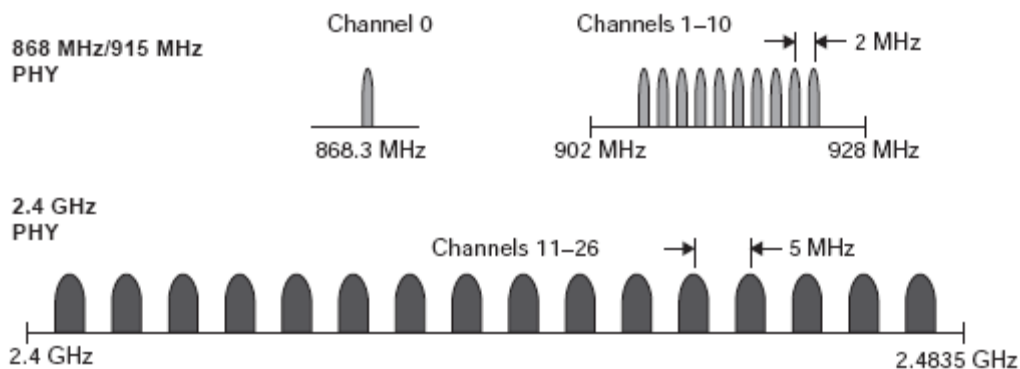
**Tabla 5.2.** Frecuencias y velocidad de transmisión en IEEE 802.15.4. Figura obtenida de [2].

IEEE 802.15.4 requiere que si un transceiver soporta la banda de 868 MHz, también debe soportar la banda de 915 MHz, y viceversa. Como se ve en la tabla 5.2, existe una especificación principal para las bandas 868/915 MHz y dos especificaciones opcionales. La especificación principal es más simple de implementar, pero las velocidades de transmisión son menores (20 Kbps y 40 Kbps respectivamente).

Antes de la introducción de estos dos modos de operación opcionales en 2006, la única manera para obtener velocidades superiores a 40 Kbps era utilizar la banda de 2.4 GHz. Ahora, con la adición de estas dos opciones, si por alguna razón no es posible operar en la banda de 2.4 GHz (por ejemplo, que exista una fuerte interferencia en la banda) o si la velocidad de 40 Kbps no es suficiente, el usuario tiene la opción de lograr 250 Kbps en las bandas 868/915 MHz.

Un transceiver que trabaja a 2.4 GHz debería soportar también las bandas de 868 y 915 MHz, aunque IEEE 802.15.4 no lo requiere.

Respecto al número de canales para cada banda frecuencial, la banda de 868 MHz sólo tiene uno, la banda de 915 MHz tiene 10 (sin contar los canales opcionales) y la banda de 2.4 GHz tiene un total de 16 canales. En la figura 5.9 se muestra la disposición de estos canales.



**Figura 5.9.** Canales IEEE 802.15.4. Figura obtenida de [6].

La banda ISM de 2.4 GHz se acepta en todo el mundo, es la que tiene la mayor tasa de transmisión de datos y el mayor número de canales. Por estas razones es la opción más popular a la hora de desarrollar transceivers. Aunque la elección de esta banda también tiene sus desventajas, como los posibles problemas de coexistencia con IEEE 802.11b, que opera a la misma frecuencia. Además, cuanto menor es la frecuencia, la señal penetra mejor en las paredes y en varios objetos, por lo que en algunas aplicaciones la opción de las bandas 868/915 MHz puede ser la mejor.

En 802.15.4 hay tres tipos de modulación: BPSK, ASK y O-QPSK. En BPSK y O-QPSK los datos viajan en la fase de la señal, mientras en ASK los datos viajan en la amplitud de la señal.

1- BPSK (Binary Phase Shift Keying):

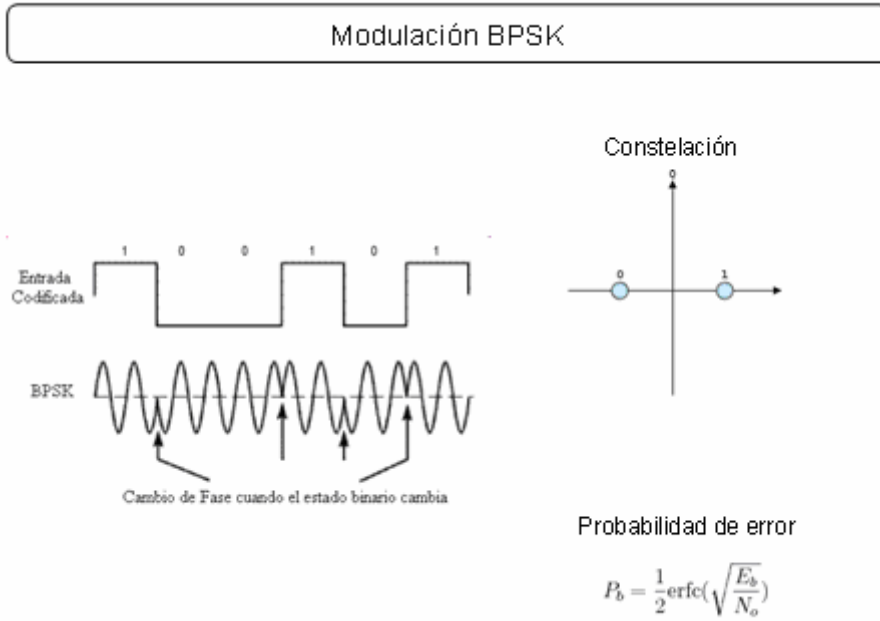


Figura 5.10. Modulación BPSK.

2- ASK (Amplitude Shift Keying):

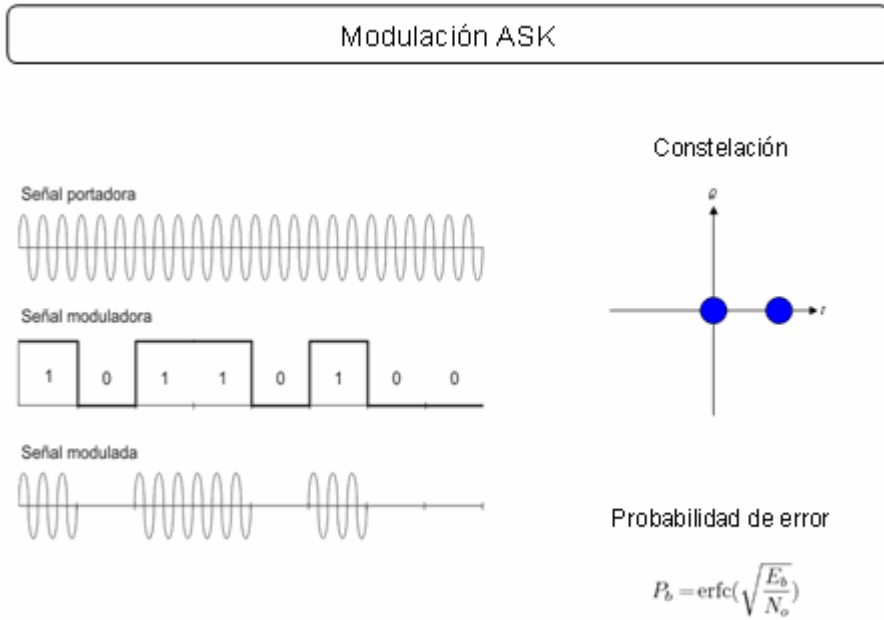
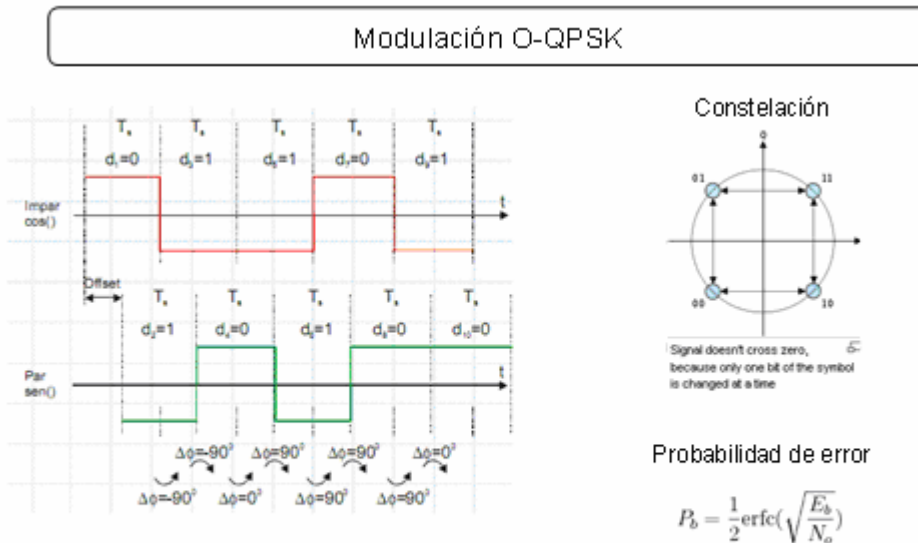


Figura 5.11. Modulación ASK.

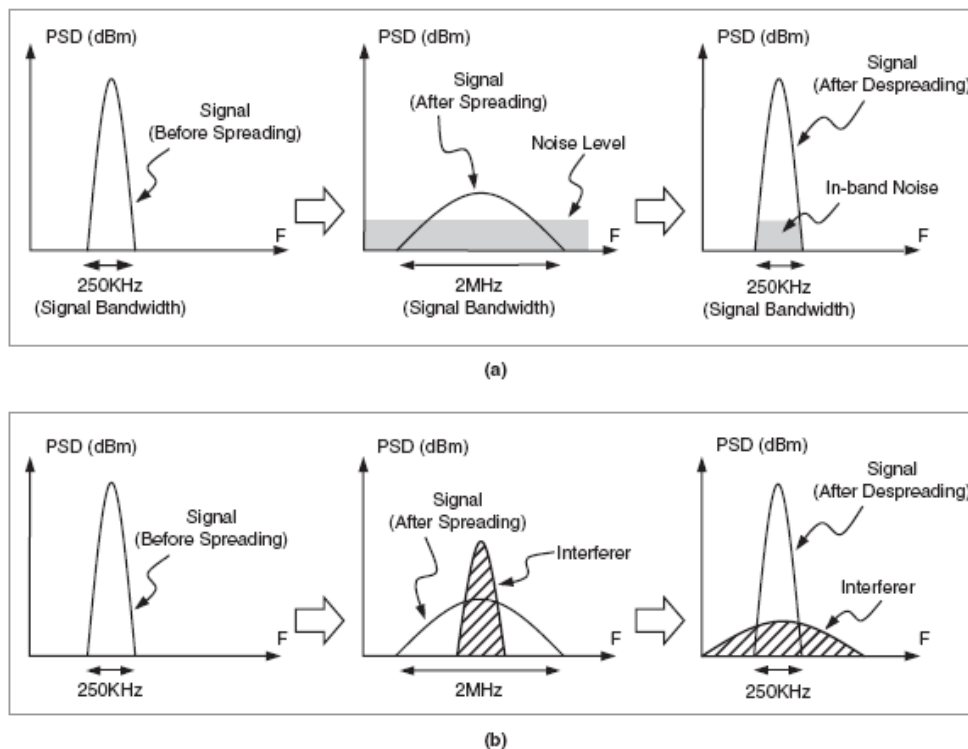
### 3- O-QPSK (Offset Quadrature Phase Shift Keying):



Al introducir un retardo de  $T_b/2$  en la componente en cuadratura, las transiciones de fase están limitadas a  $90^\circ$  y son el doble de frecuentes que en QPSK.

**Figura 5.12.** Modulación O-QPSK.

Todos los métodos de comunicación inalámbrica que aparecen en la tabla 5.2 usan las técnicas de DSSS (Direct Sequence Spread Spectrum) o PSSS (Parallel Sequence Spread Spectrum) para mejorar el rendimiento del receptor en un entorno con multitrayectos.



**Figura 5.13.** DSSS y PSSS ayudan a mejorar la SNR (a) y a reducir el efecto de interferencias (b). Figura obtenida de [2].



### 5.2.5.-Interoperabilidad

ZigBee tiene un amplio abanico de aplicaciones, y hay un gran número de empresas que desarrollan y construyen distintos tipos de soluciones ZigBee. Es muy importante que todos estos dispositivos basados en ZigBee sean capaces de interactuar entre sí sin importar cuál es su origen. Es decir, los dispositivos deben ser interoperables.

Esta interoperabilidad es una de las grandes ventajas que ofrece el protocolo ZigBee. Todos los dispositivos basados en ZigBee son capaces de comunicarse entre sí, aunque los mensajes que manden estén encriptados por cuestiones de seguridad.

### 5.2.6.-Tipos de dispositivos

Hay dos tipos de dispositivos en las redes inalámbricas IEEE 802.15.4:

1- FFD (Full Function Device): Son capaces de realizar todas las funciones que se describen en el estándar IEEE 802.15.4, y son capaces de aceptar cualquier rol en la red.

2- RFD (Reduced Function Device): Como su propio nombre indica, tienen una capacidad limitada a la hora de realizar funciones. Por ejemplo, mientras un FFD puede comunicarse con cualquier otro dispositivo de la red, los RFD sólo pueden comunicarse con un FFD. Lógicamente, la capacidad de procesamiento y de memoria de los dispositivos RFD suele ser menor que en los FFD.

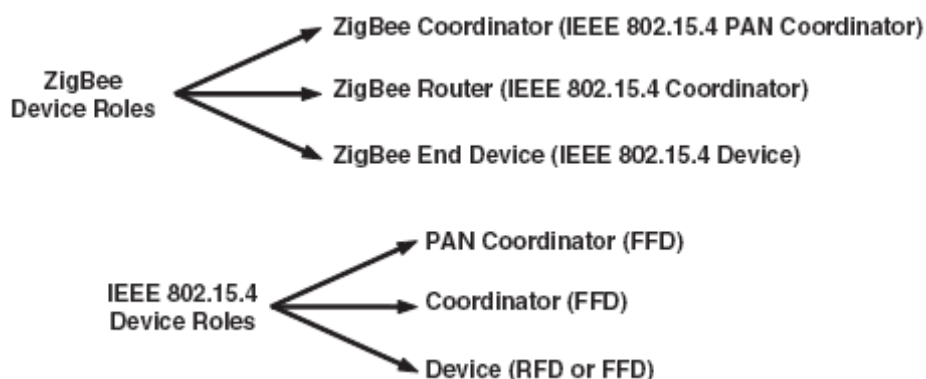
### 5.2.7.-Roles de los dispositivos

En una red IEEE 802.15.4, un dispositivo FFD puede tomar tres diferentes roles: *coordinador* (Coordinator), *coordinador PAN* (PAN coordinator), y *dispositivo* (Device). Un *coordinador* es un dispositivo FFD que es capaz de enrutar mensajes. Si el coordinador es también el principal controlador de la red, se le llama *coordinador PAN*. Si un dispositivo no está actuando como coordinador, entonces simplemente se le llama *dispositivo*.

El estándar ZigBee usa una terminología algo diferente. Un *coordinador* ZigBee (ZC) es un *coordinador PAN* de IEEE 802.15.4. Un *router* ZigBee (ZR) es un dispositivo que puede actuar como un *coordinador* de 802.15.4.



Finalmente, un *dispositivo final* ZigBee (ZED: ZigBee End Device) es un dispositivo que no es ni un *coordinador* ni un *router*. Un end device de ZigBee tiene la menor capacidad de procesado y de memoria. Normalmente suelen ser los dispositivos más baratos de la red. En la figura 5.X se puede ver un esquema con los roles que pueden tener los dispositivos tanto según el estándar ZigBee como el 802.15.4.



**Figura 5.14.** Roles de los dispositivos según el estándar ZigBee y el estándar IEEE 802.15.4. Figura obtenida de [2].




El coordinador ZigBee, ZC, es el único dispositivo que puede formar una red, por lo que cada red sólo tiene un ZC. Si la red es segura y está abierta a posibles nuevos dispositivos, el ZC debe estar presente, ya que aunque puede que no sea necesario para un funcionamiento normal de la red, se encarga de decidir si permite a un dispositivo incorporarse a ella. Si por el contrario, la red está configurada y terminada, es decir, no se espera que se unan más dispositivos, el ZC pasa a comportarse como un router ZigBee, ZR. También es el nodo raíz cuando se usa la topología de red tipo árbol.

El router ZigBee, ZR, es un componente opcional de la red. Extiende la cobertura de la red descubriendo y asociándose con ZC y ZR. Maneja la asignación de las direcciones locales y participa en el enrutamiento de los mensajes, buscando los ZED que están asociados a él.

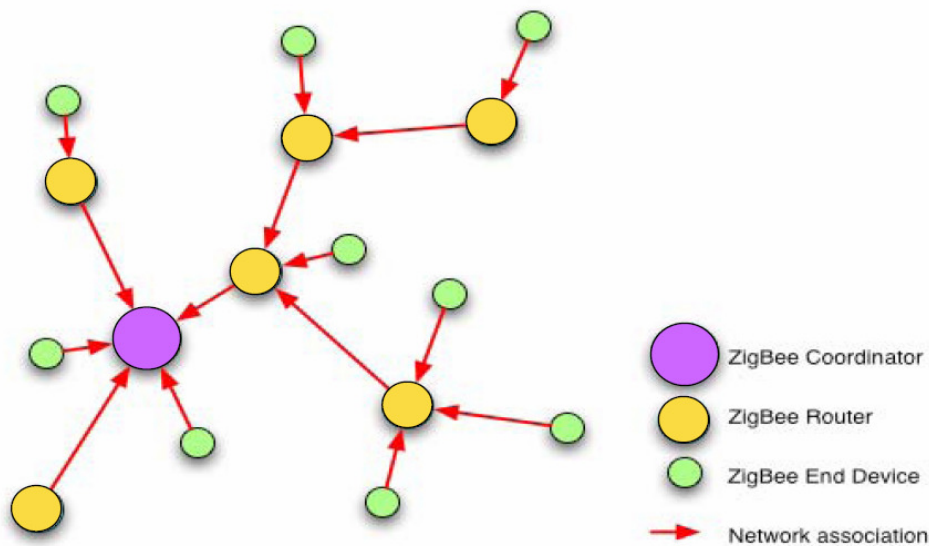
Cuando se necesita un dispositivo más simple y que cuando la red esté inactiva se pueda cambiar a estado dormido (sleep mode) para consumir menos energía, entonces se usa un ZED. El estándar ZigBee no impone tiempo límite para el estado de ahorro de energía, por lo que este tiempo dependerá de las necesidades de cada aplicación. También es un dispositivo opcional de la red, y extiende la cobertura de la red asociándose con ZC y ZR, pero no con otros ZED. No participa en el enrutamiento.

Hay que destacar que cualquier aplicación puede ser llevada a cabo por cualquiera de los tres tipos de nodos ZigBee. Por ejemplo, un ZC, ZR o ZED pueden contener una luz, un interruptor, un sensor de temperatura, etc.

En las siguientes figuras se muestran un resumen de las funciones principales de cada tipo de dispositivo ZigBee y el modo de asociación entre distintos tipos de nodos:

Node Type	Features
	Forms network Routes packets Security Trust Center Allows nodes to join network
	Joins network Routes Packets Allows nodes to join network
	Joins network Battery operated and may sleep Smallest code size Can be RxOnIdle = true (no polling) or RxOnIdle = false (polling)

**Figura 5.15.** Tipos de nodos ZigBee.



**Figura 5.16.** Asociación en una red de los distintos tipos de nodos ZigBee.



### 5.2.8.- Topologías de red

La capa de red de ZigBee es la encargada de formar y gestionar la red. La red debe estar formada en una de los dos tipos de topologías especificadas en IEEE 802.15.4: En estrella o peer-to-peer.

En la topología en estrella, mostrada en la figura 5.17, los dispositivos de la red sólo pueden comunicarse con el *coordinador PAN*. En una topología peer-to-peer, que se muestra en la figura 5.18, cada dispositivo que forma la red puede comunicarse directamente con cualquier otro dispositivo de esa misma red, siempre que la distancia entre ellos lo permita. En este caso, cualquier dispositivo FFD podría tener el rol de *coordinador PAN*. Normalmente, el *coordinador PAN* suele ser el primer dispositivo FFD que comienza a comunicarse y a formar una red. En una red peer-to-peer, todos los dispositivos que participan en el enrutamiento de los mensajes son FFD, ya que los RFD no son capaces de hacerlo.

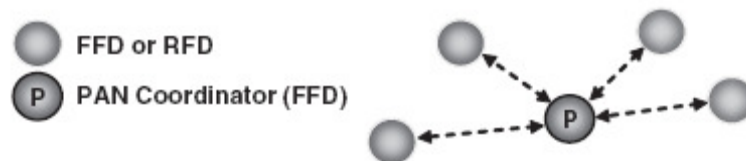


Figura 5.17. Topología de red en estrella. Figura obtenida de [2].

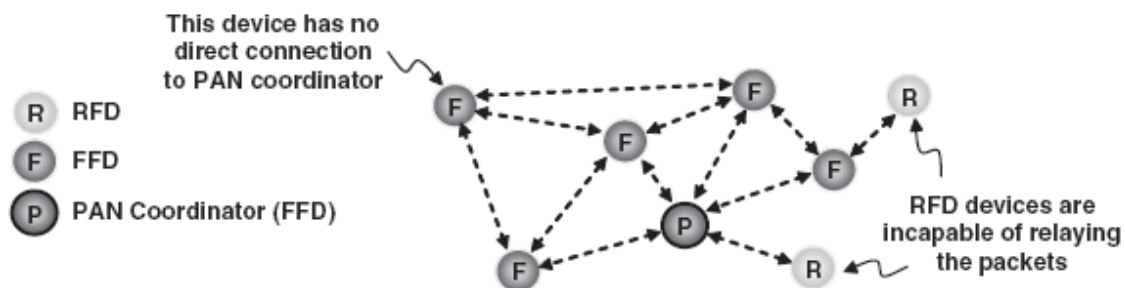
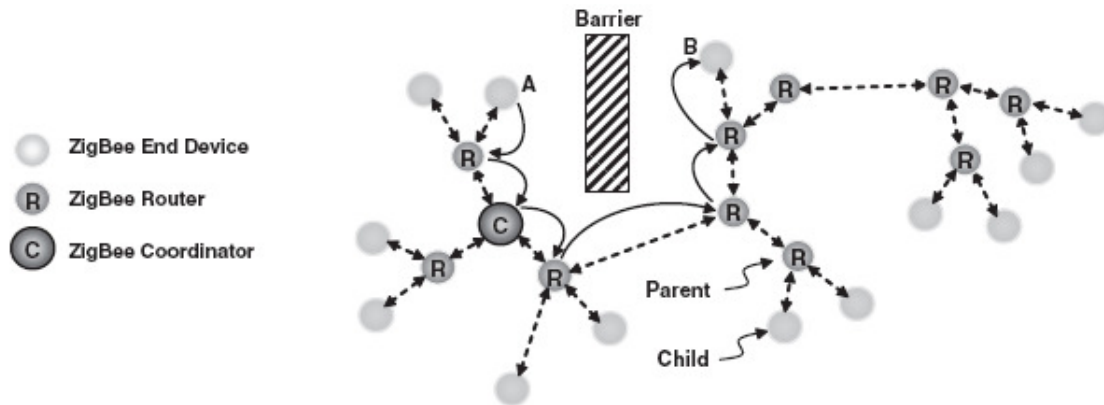


Figura 5.18. Topología de red peer-to-peer (en malla). Figura obtenida de [2].

Las redes peer-to-peer pueden tomar diferentes formas definiendo las restricciones de los dispositivos a la hora de comunicarse entre ellos. Cuando no hay restricciones, se llama topología en malla (mesh topology). Otra forma de red peer-to-peer es la topología en árbol (tree topology). En la figura 5.19 se puede ver un ejemplo de topología en árbol. En este caso, un coordinador ZigBee (PAN coordinator) crea la red inicial. Los routers ZigBee forman ramas y transmiten los mensajes hacia su destino. Los ZED actúan como hojas del árbol, no participando en el enrutamiento de los mensajes.

Los ZR pueden hacer crecer la red inicial establecida por el ZC. En la figura 5.19 se puede observar cómo los ZR ayudan a extender la red, ayudando a evitar obstáculos que dificultarían o impedirían una transmisión directa. Mediante estos saltos, llamado multihopping, el mensaje enviado por A puede llegar a su destino B.



**Figura 5.19.** Topología en árbol de ZigBee. Figura obtenida de [2].

Una red IEEE 802.15.4, independientemente de su topología, siempre será creada por un *coordinador PAN*. Como mínimo, se encargará de las siguientes tareas:

- Asignar una única dirección de 16 o 64 bits a cada dispositivo de la red.
- Mandar, recibir y enrutar mensajes a través de la red.
- Seleccionar un único identificador de PAN para su red.

Debido a las tareas que ejerce el coordinador PAN, suele estar activo durante largos períodos de tiempo, por lo que normalmente suele alimentarse con una fuente mayor que una simple batería. Los demás dispositivos suelen estar alimentados por baterías.

### 5.2.9.-Aspectos básicos de la comunicación

En este apartado se explicarán con más detalle algunos de los conceptos que se han mencionado anteriormente, como el método de acceso múltiple, los métodos de transmisión de datos o el direccionamiento en IEEE 802.15.4 y ZigBee.



### 5.2.9.1.-CSMA-CA

IEEE 802.15.4 implementa un método simple que permite a múltiples dispositivos usar el mismo canal frecuencial para comunicarse. Este método es el llamado CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance). ZigBee es asíncrono, por lo que cualquier nodo puede transmitir en cualquier instante. CSMA-CA, que es un mecanismo a nivel MAC, previene que los mensajes enviados por los nodos colisionen entre sí en el canal.

En CSMA-CA, cada vez que un dispositivo quiere transmitir, primero se asegura que el canal que va a usar no lo está siendo usado por otro dispositivo. Una vez que comprueba que el canal está libre, transmite su mensaje. Si el canal no está libre, el dispositivo espera un tiempo definido con anterioridad y lo intenta de nuevo. Este proceso se repite hasta que el canal esté libre o hasta alcanzar el máximo número de intentos definido por el usuario.

### 5.2.9.2.-Beacon-Enabled frente a Nonbeacon Networking

Los dispositivos conectados a una red tienen dos métodos para acceder al canal de transmisión: Con contención (contention based channel access) y sin contención (contention free channel access).

En el caso con contención, todos los dispositivos que quieren transmitir en el mismo canal frecuencial usan el mecanismo CSMA-CA, y el primero que se encuentra el canal libre comienza a transmitir.

En el caso sin contención, el coordinador PAN puede dedicar un espacio temporal a un dispositivo en particular. Estos espacios se llaman Guaranteed Time Slots (GTS). Un dispositivo con un GTS asignado comenzará a transmitir durante ese período de tiempo, sin usar CSMA-CA.

Para asignar un GTS, el coordinador necesita asegurarse de que todos los dispositivos de la red están sincronizados. Los mensajes llamados *Beacon*, o trama piloto, son tramas con un formato específico que se usan para sincronizar los nodos de la red. Este formato, al igual que el de los demás tipos de tramas, se explicará en el apartado 5.2.13. El coordinador PAN tiene la opción de transmitir tramas piloto para sincronizar los dispositivos. Cuando esto pasa, a la red se le llama *beacon-enabled PAN*. Es decir, la red funciona y se organiza mediante las tramas piloto que manda el coordinador. La mayor desventaja cuando se usan tramas piloto es que todos los dispositivos de la red deben salir de su modo de ahorro de energía regularmente para escuchar estos mensajes, sincronizar sus relojes y volver al modo de ahorro de energía. Por tanto, los dispositivos consumen energía solamente para sincronizarse, por lo que la vida de las baterías

de los dispositivos es mucho menor que para los dispositivos de redes que no usan tramas piloto.

Por el contrario, una red en la que el coordinador PAN no transmite tramas piloto es conocida como *nonbeacon network*. Una red de este tipo no puede usar GTS ya que no se puede sincronizar a los dispositivos. Como ya ha comentado en el párrafo anterior, la batería de los dispositivos durará más, ya que saldrán del estado de ahorro de energía muchas menos veces.

### 5.2.9.3.-Métodos de transmisión de datos

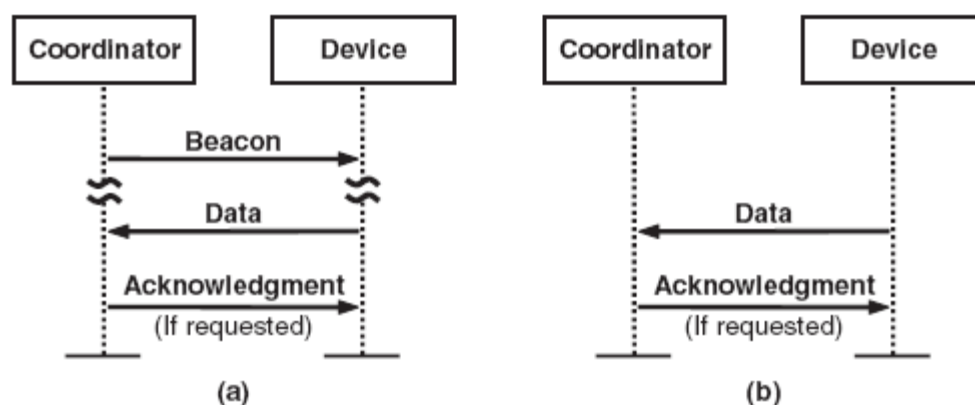
En el estándar IEEE 802.15.4 vienen definidos tres modos de transmisión de datos:

- Transmisión de datos de coordinador a dispositivo.
- Transmisión de datos de dispositivo a coordinador.
- Transmisión de datos entre iguales.

Los tres métodos pueden ser usados en topologías peer-to-peer. Pero en topologías en estrella sólo las dos primeras están permitidas.

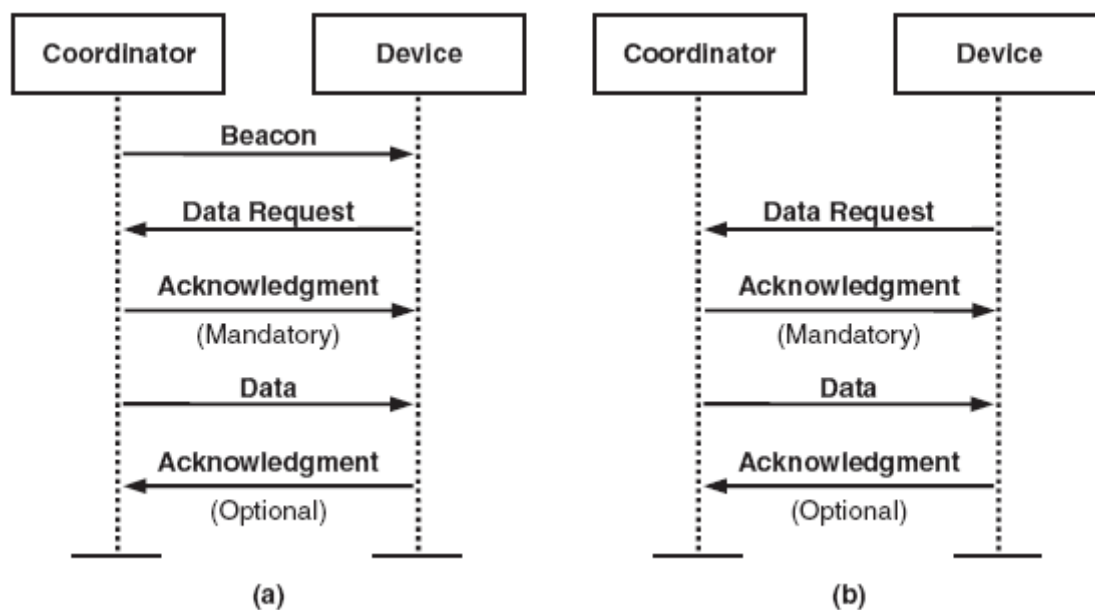
**1-Transmisión de datos de dispositivo a coordinador:** En una red con tramas piloto habilitadas, cuando un dispositivo decide enviar datos al coordinador, el dispositivo sincroniza su reloj regularmente y transmite usando CSMA-CA (suponiendo que no transmite durante un GTS). El coordinador mandará un ACK sólo si el transmisor lo requiere.

En la figura 5.20 se muestra la secuencia de mensajes de la transmisión de datos descrita en el párrafo anterior (a). También se muestra la secuencia para una red sin tramas piloto (b). En este segundo escenario, el dispositivo transmite los datos en cuanto el canal está libre. La transmisión de ACK por parte del coordinador también es opcional.



**Figura 5.20.** Transmisión de datos a un coordinador en IEEE 802.15.4: (a) con trama piloto o beacon, y (b) sin trama piloto. Figura obtenida de [2].

**2-Transmisión de datos de coordinador a dispositivo:** La figura 5.21 (a) muestra la secuencia de mensajes en una transmisión de datos de un coordinador a un dispositivo en una red con tramas piloto (beacon-enabled network). Si el coordinador necesita transmitir a un dispositivo en concreto, en la trama piloto indica que hay un mensaje pendiente para dicho dispositivo. Cuando el dispositivo recibe la trama piloto, manda una petición (data request) indicando que está listo para recibir los datos. El coordinador manda un ACK para indicar que ha recibido la petición, y a continuación transmite los datos. El envío de ACK por parte del dispositivo es opcional.



**Figura 5.21.** Transmisión de datos de un coordinador en IEEE 802.15.4: (a) con trama piloto o beacon, y (b) sin trama piloto. Figura obtenida de [2].

En una comunicación sin trama piloto (figura 5.21b), para comenzar una transmisión, el coordinador debe esperar a que le llegue una petición de datos desde un dispositivo. Si un dispositivo manda una petición pero no existen datos pendientes para enviar a dicho dispositivo, el coordinador envía un ACK con un formato específico para indicar que no hay datos para este dispositivo.

**2-Transmisión de datos entre iguales (peer-to-peer):** En una topología peer-to-peer, cada dispositivo puede comunicarse directamente con cualquier otro.

Hay que destacar que en ZigBee existen varias opciones para el envío de mensajes. A continuación se hará una breve explicación de cada una de estas opciones:



-Unicast sin ACK: La transmisión se da sólo entre un nodo transmisor y un nodo receptor. A no se que los dos nodos sean vecinos y puedan comunicarse directamente (que estén dentro del rango de transmisión radio), el descubrimiento de la ruta tiene lugar la primera vez que los dos nodos se comunican entre sí.

-Unicast con ACK: Es igual que unicast sin ACK, pero en la transmisión se requiere que el nodo destino transmita un ACK para confirmar que ha recibido el mensaje.

-Broadcast: El mensaje es transmitido por un nodo a todos los demás nodos de la red.

-Groupcast/Multicast: Es como broadcast, pero a un grupo específico de nodos. Los nodos que reciban el mensaje y no sean del grupo al que va dirigido, lo descartan.

#### 5.2.9.4.-Verificación de los datos

Un paquete es un número de bits transmitidos conjuntamente en un formato específico. El receptor necesita tener un mecanismo para verificar que todos los bits recibidos se recuperan sin errores. IEEE 802.15.4, para detectar estos posibles errores en el paquete de datos, usa un mecanismo basado en un código cíclico CRC (Cyclic Redundancy Check) llamado FCS (Frame Check Sequence) de 16 bits.

#### 5.2.9.5.-Direccionamiento

Cada dispositivo perteneciente a una red necesita una única dirección. IEEE 802.15.4 usa dos métodos de direccionamiento, uno de direccionamiento corto de 16 bits y otro de direccionamiento extendido de 64 bits. Cada red puede elegir cualquiera de los dos métodos. Lógicamente, usar el método con direccionamiento corto reduce el tamaño de los mensajes y el espacio de memoria necesario para almacenar las direcciones. Pero el uso de las direcciones de 64 bits hace que el número de dispositivos que se pueden conectar a una red sea casi ilimitado

En ZigBee, los nodos sólo pueden comunicarse con nodos de su misma red PAN. Como ya se ha comentado anteriormente en este trabajo, las PAN las forman los coordinadores (ZC). Los demás nodos, ya sean ZR o ZED, pueden unirse a las redes, pero nunca formar una ellos mismos.

Para poder diferenciar los nodos pertenecientes a cada red ZigBee, existen los denominados PAN IDs (Personal Area Network identifiers). Este



identificador se hace imprescindible en los casos en que haya nodos de otras redes muy cerca, sobre todo si comparten el mismo canal frecuencial. Así se asegura la coexistencia de varias redes que estén próximas entre sí.

Los PAN IDs usan direcciones cortas, de 16 bits, que van desde 0x0000 a 0x3fff. El rango varía del que especifica 802.15.4, que va desde 0x0000 a 0xffff. Generalmente, las redes ZigBee no entran en conflicto, ya que es prácticamente imposible que existan 16.000 diferentes redes en el mismo rango radio.

ZigBee usa las PAN IDs de 16 bits para todas sus comunicaciones, excepto para una. Cuando un nodo quiere unirse a una red específica, se usan los identificadores EPIDs (Extended PAN IDs). Estos identificadores son direcciones de 64 bits que identifican inequívocamente una PAN. Así se evitan posibles errores (que existan dos redes relativamente cercanas con el mismo PAN IDs), y el nodo se incorporará exactamente a la red correcta.

Es interesante destacar que los EPIDs no tienen ninguna relación con los PAN IDs. Son direcciones únicas de 64 bits que se usan solamente para que un nodo encuentre la red correcta.

Además de identificar las redes, también es necesario identificar cada uno de los nodos que pertenecen a la red. Para ello se usa una dirección llamada dirección de red (NwkAddr), también llamada dirección corta o dirección de nodo. NwkAddr es una dirección de 16 bits que identifica a los nodos de una red. El coordinador ZigBee siempre tiene como dirección 0x0000.

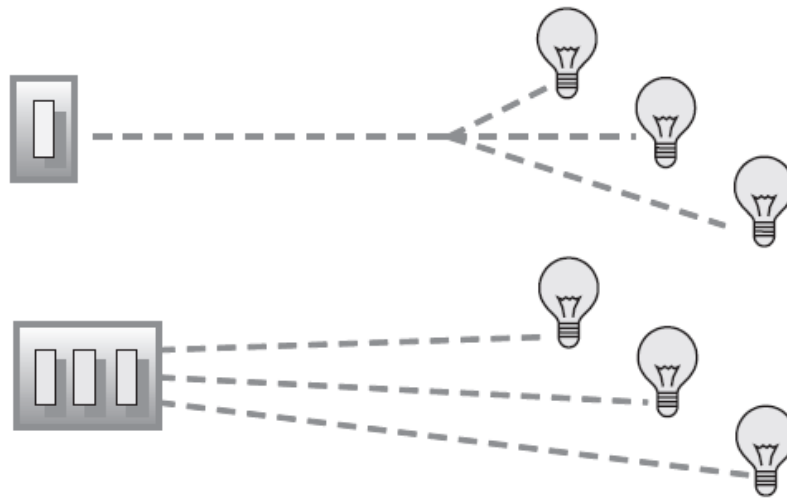
### [5.2.10.-Asociación y disasociación](#)

La asociación y disasociación son servicios que proporciona el estándar IEEE 802.15.4 para que los dispositivos puedan unirse o dejar una red. Cuando un dispositivo quiere unirse a una red, manda una petición de asociación al coordinador de esa red. Entonces, el coordinador podrá aceptar o rechazar esa petición. En el caso de disasociación, el dispositivo que quiere dejar la red manda una petición al coordinador, indicándole que quiere dejar la red. La decisión también será tomada por el coordinador.

### [5.2.11.-Vinculación de dispositivos \(Binding\)](#)

A la tarea de crear enlaces lógicos entre aplicaciones que están relacionadas se le llama vinculación de dispositivos o *binding*. Por ejemplo, un dispositivo ZigBee conectado a una lámpara podría ser ventajoso que estuviera

relacionado lógicamente con el dispositivo ZigBee conectado al interruptor que controla la lámpara (Figura 5.22).



**Figura 5.22.** Mediante el *bindig* se pueden vincular lógicamente dos o más dispositivos ZigBee. Figura obtenida de [1].

La información de estos enlaces lógicos entre dispositivos de una red se almacena en una tabla, llamada *binding table*. El estándar ZigBee, a nivel de la capa de aplicación, da soporte a la creación y mantenimiento de estas tablas.

### [5.2.12.-Funciones y estructura de trama de las distintas capas del estándar IEEE 802.15.4 y ZigBee](#)

Este apartado podría haber sido parte del apartado 5.2.3, donde se comenta la relación entre el estándar 802.15.4 y ZigBee y se comentan las capas que define cada uno. Pero se decidió colocarlo al final del apartado 5.2, ya que así será más fácil entender, puesto que anteriormente ya han sido nombrados y explicados varios conceptos, como por ejemplo, la función de algunos tipos de trama, la asociación y disasociación, etc.

#### [5.2.12.1.-Capa física \(PHY\)](#)

Como se vio en la figura 5.4, y que a continuación se muestra de nuevo, para el protocolo total de ZigBee, la capa más baja es la capa física del estándar IEEE 802.15.4. Esta capa es la más cercana al hardware, y es la que controla y comunica directamente con el transceiver radio.



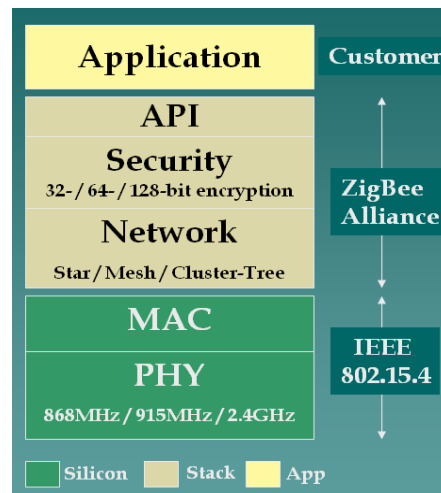


Figura 5.23. ZigBee y el estándar IEEE 802.15.4

La capa física, PHY, es la responsable de activar la comunicación radio con la que se transmitirán o recibirán los paquetes. Esta capa también es la encargada de seleccionar el canal frecuencial y asegurarse de que el canal no está siendo usado por otro dispositivo.

### 5.2.12.1.1-Estructura general del paquete PHY:

Los datos y comandos que envían los dispositivos se transmiten en forma de paquete. La figura 5.24 muestra la estructura general de estos paquetes. El primer bit transmitido es el bit menos significativo (LSB) de la cabecera SHR. El bit más significativo del último octeto de la carga de la capa física es transmitido en último lugar.

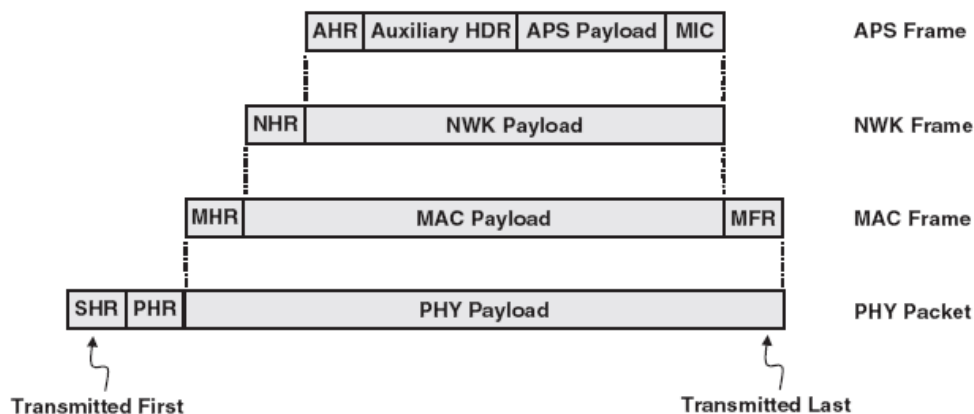


Figura 5.24. Estructura de paquete ZigBee. Figura obtenida de [2].

El paquete de la capa física se compone de tres partes: La cabecera de sincronización SHR (Synchronization header), la cabecera de la capa física PHR (PHY header) y el resto de datos provenientes de las capas superiores (PHY payload). La SHR permite al receptor sincronizarse e introducirse en la cadena de bits, y la PHR contiene información sobre la longitud de la información.



También existe un campo llamado SDF, que simplemente indica el final de SHR y el comienzo de PHR. Este campo puede verse en la figura 5.25.

### 5.2.12.2.-Capa MAC

La capa de control de acceso al medio ejerce de interfaz entre la capa física y la capa de red. Es la responsable de generar tramas piloto (beacon) en el caso de que la red lo necesite. Esta capa también es la que proporciona los servicios de asociación y disasociación.

La trama MAC, que se transmite a otros dispositivos como la carga de la capa PHY, consta de tres secciones (Figura 5.24). La cabecera MHR (MAC header) contiene la información referente al direccionamiento, al tipo de trama y a la seguridad. La carga de la capa MAC (MAC payload) es la información recibida de capas superiores, su longitud es variable (incluida la longitud = 0) y contiene datos o comandos. Por último, la cola MFR (MAC footer) contiene una secuencia de 16 bits para la verificación de los datos.

IEEE 802.15.4 define cuatro estructuras diferentes de tramas MAC:

- Trama piloto (Beacon frame).
- Trama de datos.
- Trama ACK.
- Trama de comando MAC.

#### 5.2.12.2.1.-Trama piloto:

La estructura de la trama piloto puede verse en la figura 5.24. Lo que en la figura se llama PSDU (PHY Service Data Unit), es la carga de la capa física, o lo que es lo mismo, la trama entera que recibe de la capa MAC.

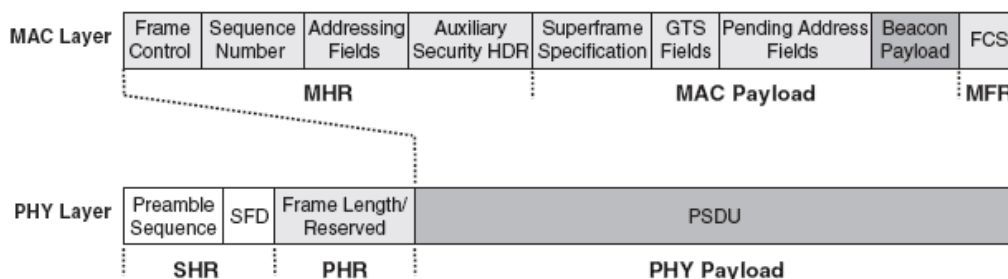


Figura 5.25. Estructura de trama piloto de la capa MAC. Figura obtenida de [2].

La diferencia entre los tipos de tramas MAC es la información que le llega de la capa de red (NWK), lo que se llama *MAC payload*. Estos campos son diferentes para cada una de las distintas tramas MAC.

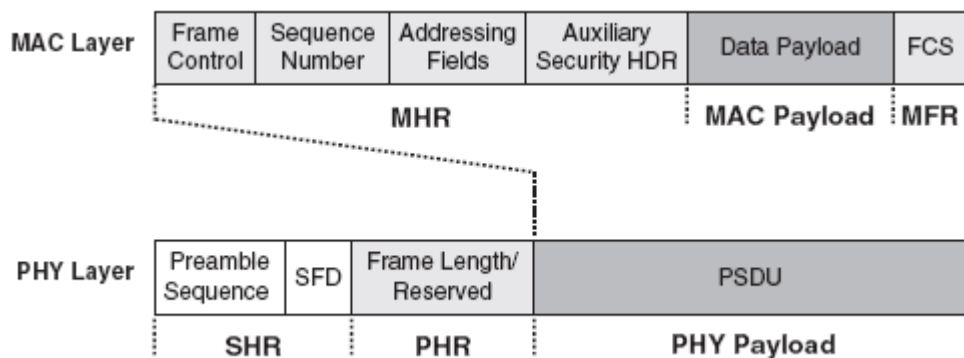


La trama piloto no sólo se usa para sincronizar los dispositivos de la red. También sirve para que el coordinador comunique a un dispositivo concreto si tiene información dirigida a él pendiente de enviar. Entonces, el dispositivo enviará una petición de datos al coordinador. Este proceso se llama transmisión indirecta.

En el campo de dirección pendiente (Pending Address) es donde se guarda la dirección del dispositivo que tiene datos pendientes de ser enviados. Cada vez que un dispositivo recibe una trama piloto, comprobará el campo de dirección pendiente para saber si el coordinador tiene información pendiente para él.

#### 5.2.12.2.2.-Trama de datos:

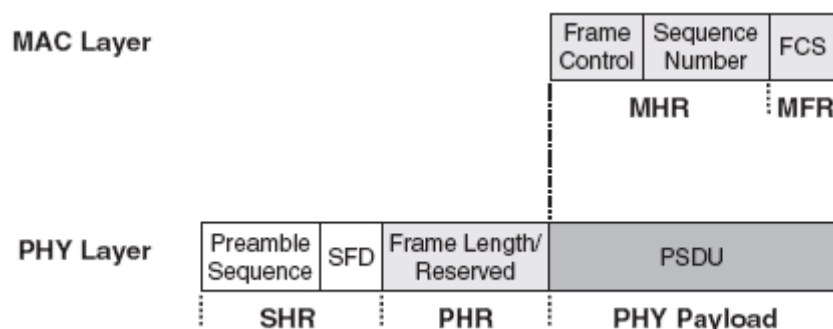
Como se muestra en la figura 5.26, esta trama transmite datos.



**Figura 5.26.** Estructura de trama de datos de la capa MAC. Figura obtenida de [2].

#### 5.2.12.2.3.-Trama ACK:

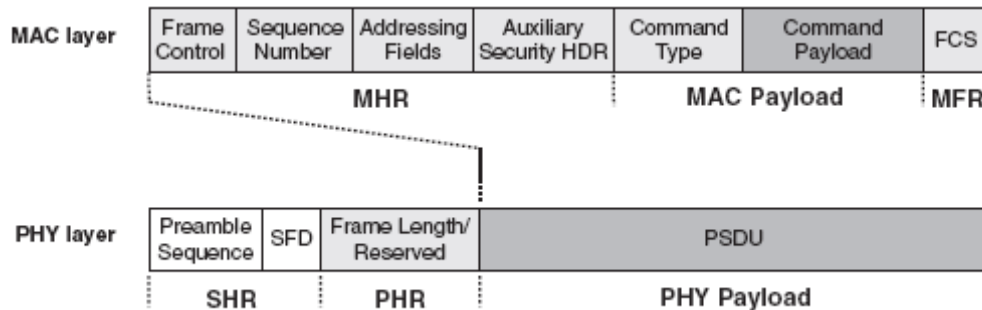
Las tramas ACK las envía un dispositivo a otro para confirmar la correcta recepción de un paquete. Son las tramas más simples de todas las tramas MAC, y como se ve en la figura 5.27, no llevan carga de la capa de red. También se reducen los campos de la cabecera MHR.



**Figura 5.27.** Estructura de trama ACK de la capa MAC. Figura obtenida de [2].

#### 5.2.12.2.4.-Trama de comando MAC:

Estas tramas transmiten los comandos MAC, como por ejemplo la asociación, la disasociación, o las peticiones de datos.



**Figura 5.28.** Estructura de trama de comando de la capa MAC. Figura obtenida de [2].

Para obtener información más concreta sobre los distintos campos de las tramas, pueden dirigirse a los documentos y libros que aparecen en la bibliografía de este trabajo.

#### 5.2.12.3.-Capa de red (NWK)

La capa de red se encuentra situada entre la capa MAC y la capa de aplicación, y es responsable de gestionar la formación y enrutamiento de la red. La capa de red de un coordinador ZigBee también es responsable de establecer una nueva red cuando la situación lo requiera, así como de seleccionar la topología de la red (árbol, estrella o malla). También se encarga de asignar las direcciones de red o *NWK addresses* a todos los dispositivos de su red.

Volviendo a la figura 5.24, la capa de red consta de dos partes: La cabecera NHR (NWK header), que contiene la información acerca del direccionamiento a nivel de red y la información de control; y en segundo lugar, la carga de la capa de red (NWK payload), que es la información que se recibe de la capa de aplicación, o la subcapa de aplicación (APS).

#### 5.2.12.4.-Capa de aplicación

Es la capa más alta en el protocolo ZigBee. Las empresas son las encargadas de crear y desarrollar sus propias aplicaciones, aunque el estándar ZigBee ofrece la opción de usar perfiles de aplicación para desarrollarlas. El uso de estos perfiles de aplicación ofrece mayor interoperabilidad entre productos desarrollados por diferentes empresas.



### 5.2.12.5.-Seguridad

En una red inalámbrica, el mensaje transmitido puede ser recibido fácilmente por un intruso. Por ello, existen dos principales motivos de preocupación respecto a la seguridad de la información transmitida.

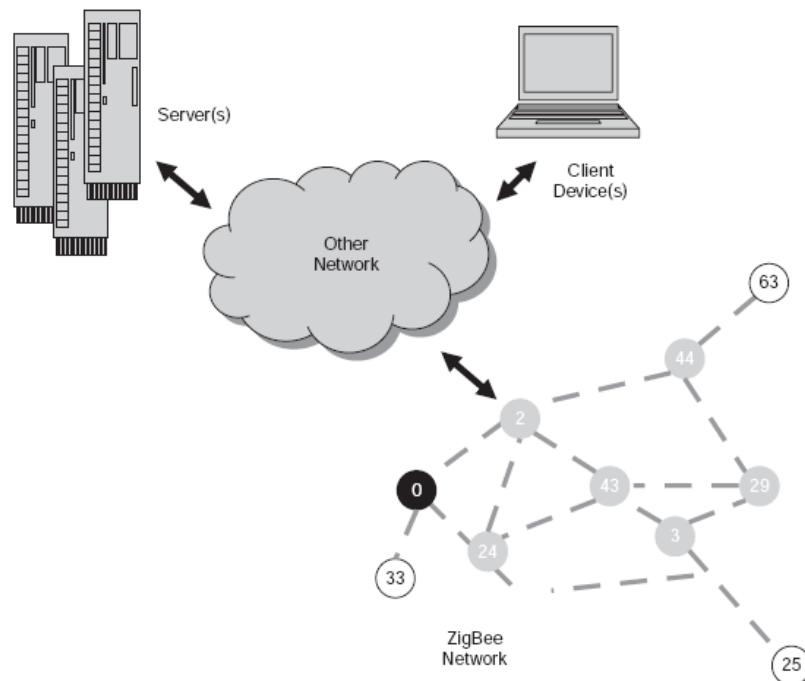
El primero es la confidencialidad. El intruso podría obtener información sensible simplemente escuchando los mensajes transmitidos. Encriptando los paquetes antes de transmitirlos se soluciona este problema. El estándar IEEE 802.15.4 soporta el uso del estándar de encriptación avanzada AES (Advanced Encryption Standard).

El segundo motivo de preocupación es que un intruso podría modificar y reenviar un paquete aunque este esté encriptado. Para evitarlo, se añade un código de integridad de mensaje (MIC) en cada paquete enviado, por lo que el receptor sabrá si un paquete ha sido modificado durante la transmisión. A este proceso se le llama autenticación.

Uno de los mayores problemas que tiene ZigBee a la hora de implementar las soluciones de seguridad es que las redes inalámbricas basadas en ZigBee tienen recursos limitados, como la limitación computacional y de memoria. Además, al ser una solución de bajo coste, el hardware de los nodos no suele estar muy protegido, por lo que un intruso podría acceder a él relativamente fácil, pudiendo obtener información directamente de la memoria del dispositivo. Aunque existen soluciones que al detectar el nodo que está siendo manipulado, borra la información sensible.

### 5.2.13.-La pasarela ZigBee

Muchas redes ZigBee conviven con otro tipo de redes, y en muchas ocasiones deben interactuar entre ellas. Una pasarela de ZigBee (ZigBee gateway) ofrece una interfaz entre una red ZigBee y otra red con otro estándar diferente (por ejemplo, Internet) para que elementos de las diferentes redes puedan comunicarse entre sí. La pasarela implementa los diferentes estándares para poder traducir los paquetes ZigBee al formato de los demás protocolos, y viceversa.



**Figura 5.29.** Pasarela de ZigBee. Figura obtenida de [1].

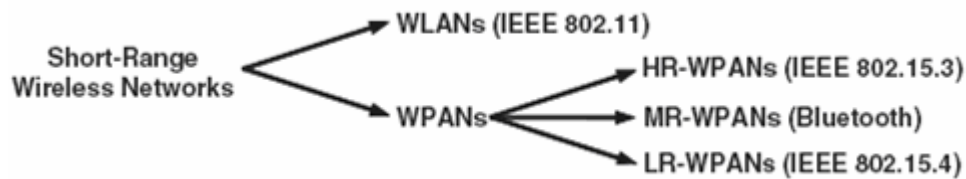
#### 5.2.14.-ZigBee frente a otras tecnologías inalámbricas

Para terminar con el apartado relacionado con ZigBee, a continuación se va a realizar una breve comparación entre ZigBee y otras tecnologías inalámbricas existentes.

Las redes inalámbricas de corto alcance pueden dividirse en dos categorías principales: Las redes inalámbricas de área local (WLAN) y las de área personal (WPAN).

Las WLAN pueden reemplazar o ser una extensión de las redes de área local (LAN) como por ejemplo Ethernet (IEEE 802.3). Generalmente, el objetivo de las WLAN es maximizar el alcance y la velocidad de transmisión.

Las WPAN, por el contrario, no se desarrollan para reemplazar ninguna red LAN. El objetivo de estas es ofrecer una solución inalámbrica de bajo consumo para lo que se denomina área personal, sin necesidad de ninguna o muy poca infraestructura. En la figura 5.30 se puede ver las clases en las que se dividen estas redes: alta velocidad o HR (High-rate), media velocidad o MR (Medium-rate) y baja velocidad o LR (Low-rate).



**Figura 5.30.** Clasificación de red inalámbricas de corto alcance. Figura obtenida de [2].

Como ya se ha comentado durante este trabajo, ZigBee es una buena solución para redes de sensores, pero la parte radio definida por 802.15.4 no es la única que ocupa este espacio, cada una con sus ventajas y desventajas respecto a cada aplicación en particular. A continuación se describen brevemente las más importantes:

-*Wireless USB*. Es una tecnología de dispositivos pequeños que operan con batería, como por ejemplo, un ratón inalámbrico. Además, no es una tecnología cara y no consume mucha batería. Aunque no tiene el alcance ni la seguridad que ofrece ZigBee. Wireless USB se basa en Ultra Wideband (UWB), que ofrece una velocidad de hasta 480 Mbps en una distancia de 2 metros. Si la velocidad se baja a 110 Mbps, la distancia que alcanza puede llegar hasta 10 metros.

-*WiFi (802.11)* es algo más cara que ZigBee y típicamente necesita mayor capacidad de procesamiento. El consumo también es mucho mayor.

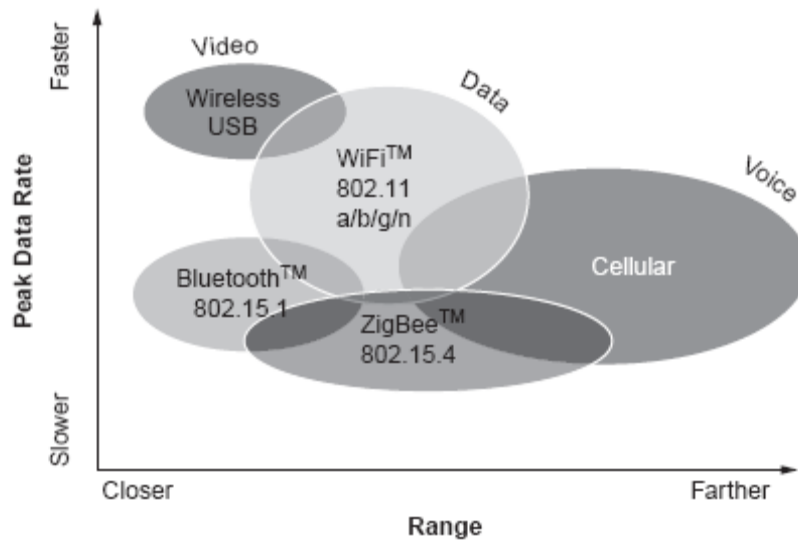
-*Bluetooth* es muy seguro y relativamente barato, pero debido a su tecnología de salto en frecuencia el consumo de energía es muy alto. Típicamente el número de dispositivos que se pueden conectar a la misma red está limitado a 7, mientras en ZigBee pueden ser miles.

-*Wibree* entra en el grupo de interés de Bluetooth, y está diseñado para aplicaciones en relojes y sensores corporales. Se diseñó para consumir muy poca potencia, pero al igual que Bluetooth, el número de nodos en una red está muy limitado.

-*Z-Wave* es una tecnología creada específicamente para aplicaciones de automatización del hogar. Compite directamente con ZigBee.

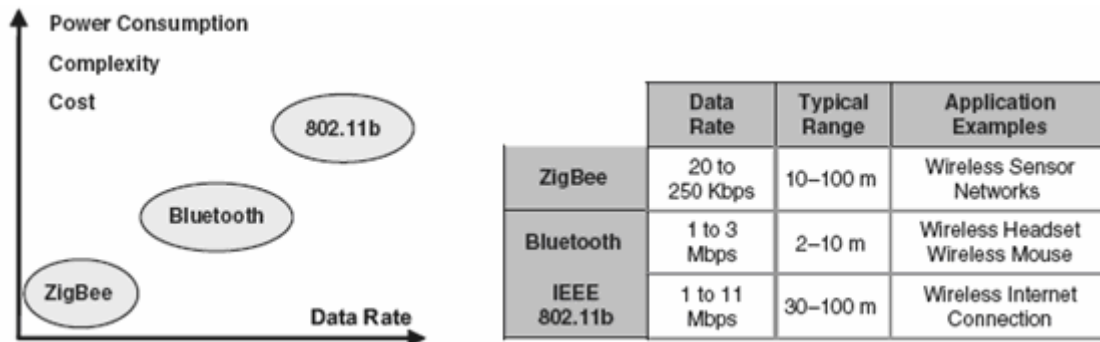
-*Soluciones propietarias*. Hoy en día son el principal competidor de ZigBee. Existen muchas soluciones buenas para varias frecuencias, pero el mayor problema que tienen es la falta de protocolo. Crear tu propia solución puede suponer un gran esfuerzo de desarrollo, o lo que es lo mismo, tiempo y dinero.

En la siguiente figura se muestra el espacio que ocupan las tecnologías mencionadas más importantes:



**Figura 5.31.** Comparación de tecnologías inalámbricas. Figura obtenida de [1].

Comparando un poco más profundamente ZigBee con Bluetooth y el estándar 802.11 (WiFi) ayudará a entender un poco mejor cómo ZigBee ocupa un lugar que los estándares más populares no ocupaban. La figura 5.32 resume las características básicas de estos tres estándares.



**Figura 5.32.** Comparación de ZigBee con Bluetooth y 802.11b. Figura obtenida de [2].

De la familia de estándares IEEE 802.11 se ha elegido el 802.11b para hacer la comparación porque opera a la frecuencia de 2.4 GHz, que es la misma a la que operan los otros dos.

En la siguiente tabla se muestran más datos para la comparación de los tres estándares:





Feature(s)	IEEE 802.11b	Bluetooth	ZigBee
Power Profile	Hours	Days	Years
Complexity	Very Complex	Complex	Simple
Nodes/Master	32	7	64000
Latency	Enumeration upto 3 seconds	Enumeration upto 10 seconds	Enumeration 30ms
Range	100 m	10m	70m-300m
Extendability	Roaming possible	No	YES
Data Rate	11Mbps	1Mbps	250Kbps
Security	Authentication Service Set ID (SSID)	64 bit, 128 bit	128 bit AES and Application Layer user defined

Tabla 5.3. Comparación de ZigBee con Bluetooth y 802.11b.

Un parámetro muy importante que no se ha comentado anteriormente es la relación señal-ruido. En la figura siguiente se representa la tasa de error de datos frente a la relación señal-ruido. Las buenas prestaciones que ofrece 802.15.4 hace que ZigBee sea mucho más robusto frente al ruido que los demás estándares.

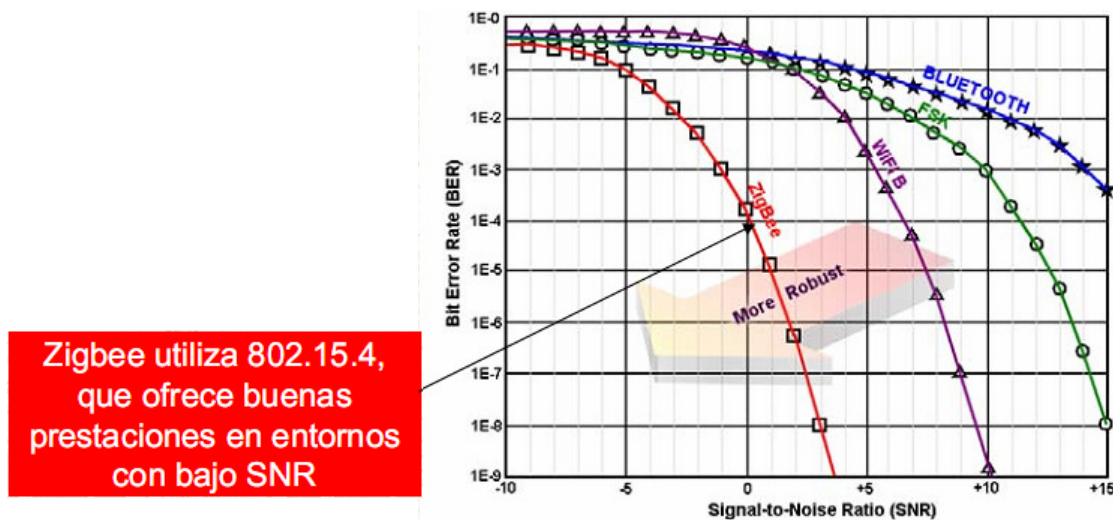


Figura 5.33. Comparación de ZigBee con Bluetooth y 802.11b.

## 6.-Dispositivos inalámbricos utilizados en el estudio

Antes de comenzar a mostrar todos los datos obtenidos en los diferentes escenarios que se han estudiado en este trabajo, en este apartado se van a mostrar y describir los dos modelos diferentes de dispositivos inalámbricos que han sido utilizados.

El primer dispositivo en ser utilizado, que también fue con el que se realizaron la mayor parte de los estudios en los distintos escenarios fue el módulo de radiofrecuencia basado en 802.15.4 XBee-Pro de Digi International. La elección de estos dispositivos fue simplemente debida al hecho de que ya existían varios de ellos en la UPNA para poder ser estudiados.

El segundo y último dispositivo que se utilizó fue un dispositivo diseñado y desarrollado por el departamento de Ingeniería Electrónica de la Universidad de Valencia, de los cuales enviaron varios para que la parte radio fuera analizada en la UPNA.

### 6.1.- XBee-Pro

El XBee-Pro es un módulo de radiofrecuencia basado en IEEE 802.15.4 de la compañía Digi International. Fueron diseñados para redes de sensores de bajo coste y bajo consumo. A continuación se van a describir las características más relevantes de estos dispositivos. Para obtener más datos, pueden dirigirse a [www.digi.com](http://www.digi.com).



**Figura 6.1.** Módulo XBee-Pro.

Las características de estos dispositivos más significativas para este trabajo pueden verse en la tabla 6.1, y son las siguientes:

- Frecuencia de operación: Banda ISM 2.4 GHz.
- Alcance en entornos interiores o urbanos: 300 metros.
- Alcance en entornos exteriores con línea de visión despejada: 1.600 m.

- Potencia transmitida máxima: 18 dBm (63 mW).
- Sensibilidad: -100 dBm.
- Máxima tasa de datos: 250 Kbps.
- Posibilidad de usar retransmisiones y ACKs.
- Comunicación unicast y broadcast.
- Modo de comunicación Non-beacon (sin tramas piloto).
- Soporta topologías en malla, árbol y estrella.
- Bajo consumo de energía:
  - Pico de corriente en transmisión: 250 mA.
  - Mínimo consumo de corriente en recepción (Idle): 55 mA.
- Están optimizados para su uso en EEUU, Canadá, Australia, Japón y Europa.
- Se suministra gratis el software de configuración X-CTU. En la siguiente figura se muestra una captura de pantalla del X-CTU.

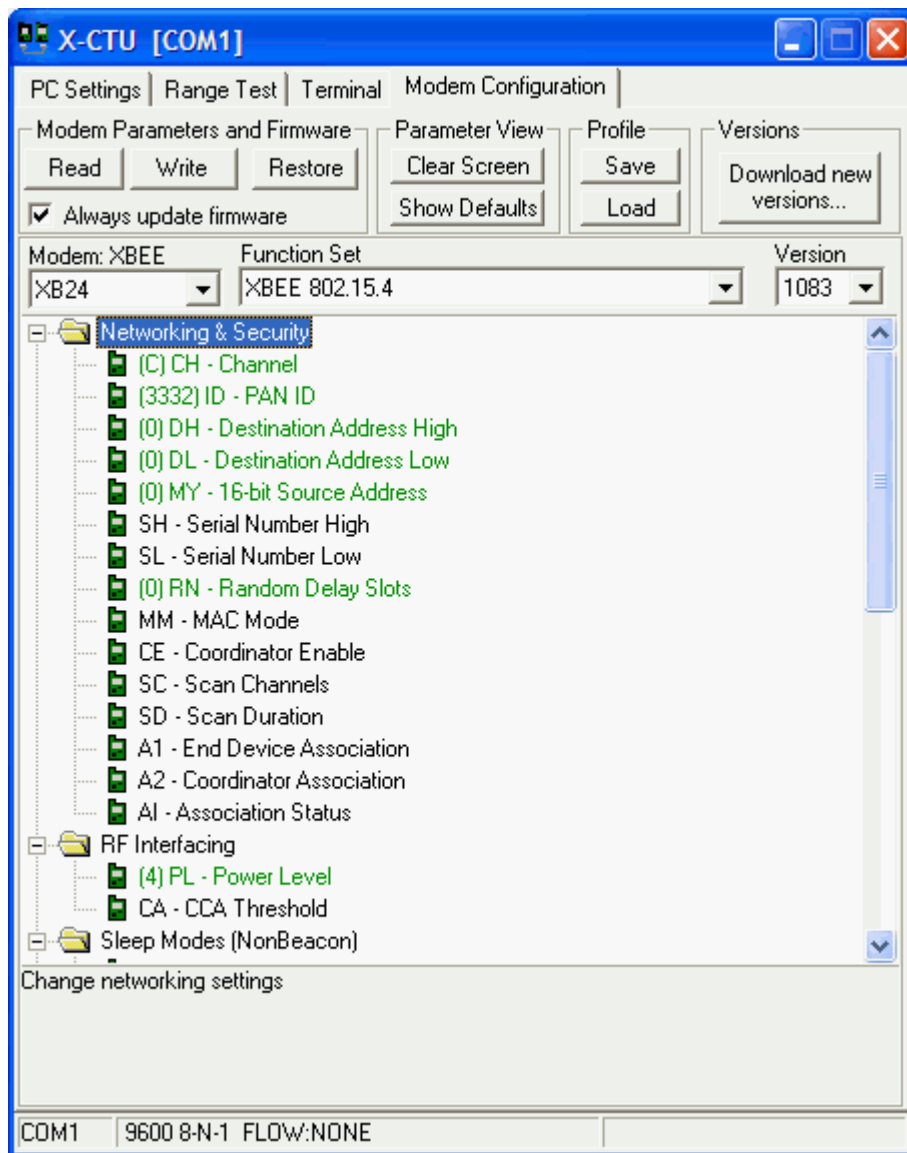


Figura 6.2. Software X-CTU.

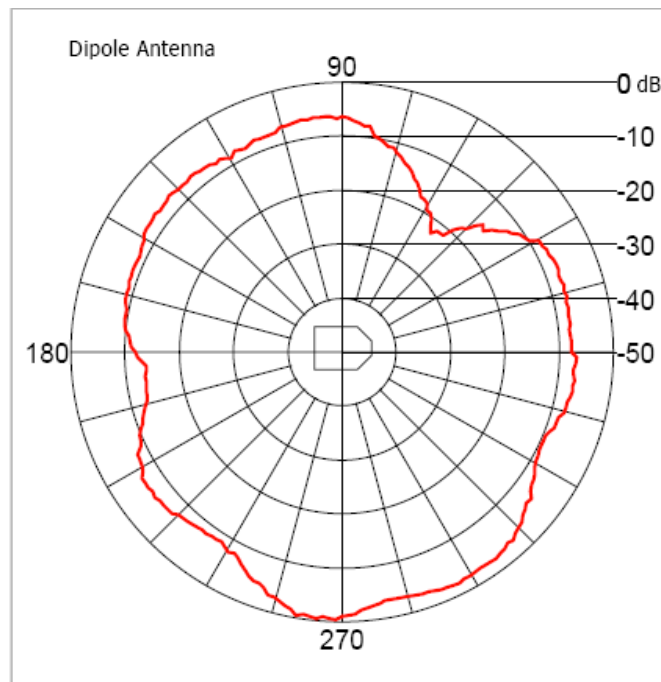


Specification	XBee-PRO
<b>Performance</b>	
Indoor/Urban Range	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-100 dBm (1% packet error rate)
<b>Power Requirements</b>	
Supply Voltage	2.8 – 3.4 V
Transmit Current (typical)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	55mA (@ 3.3 V)
Power-down Current	< 10 $\mu$ A
<b>General</b>	
Operating Frequency	ISM 2.4 GHz
Dimensions	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
<b>Networking &amp; Security</b>	
Supported Network Topologies	
Number of Channels (software selectable)	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses
<b>Agency Approvals</b>	
United States (FCC Part 15.247)	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEEPRO
Europe (CE)	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick

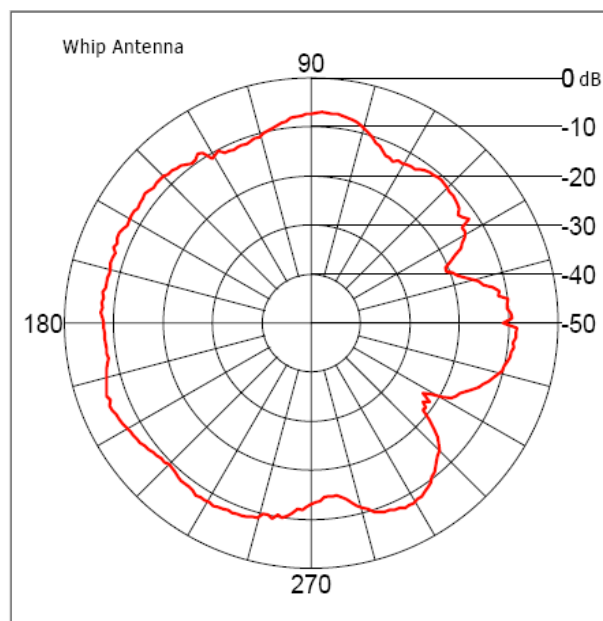
**Tabla 6.1.** Especificaciones del módulo Xbee-Pro. Tabla obtenida de [11].

Con respecto a las antenas, hay tres tipos que han sido testeadas y aprobadas para usar con estos módulos: Dipolo (2.1 dBi), Whip-monopolo (1.5 dBi) y Chip (-1.5 dBi).

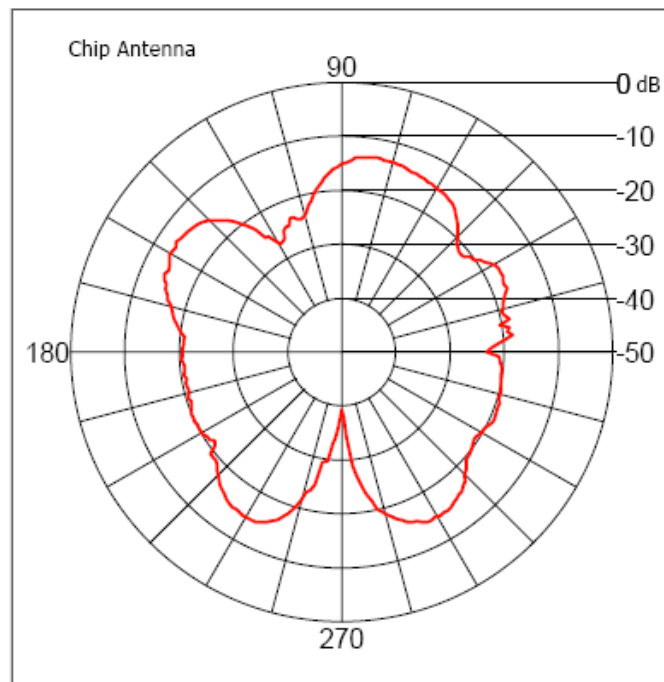
La antena que tienen nuestros dispositivos es la de Chip, que tiene una ganancia negativa (-1.5 dBi) y, como se ve en el diagrama de radiación de la figura 6.5, es una antena bastante deficiente. Es la peor antena de las tres posibles, por lo que todas nuestras medidas son el peor caso, y los resultados con las otras antenas se presuponen serán mejores que los obtenidos en este trabajo. Los siguientes diagramas de radiación están normalizados con el pico de potencia de la antena dipolo, para una mejor comparación.



**Figura 6.3.** Diagrama de radiación de la antena dipolo del módulo Xbee-Pro. Figura obtenida de [11].

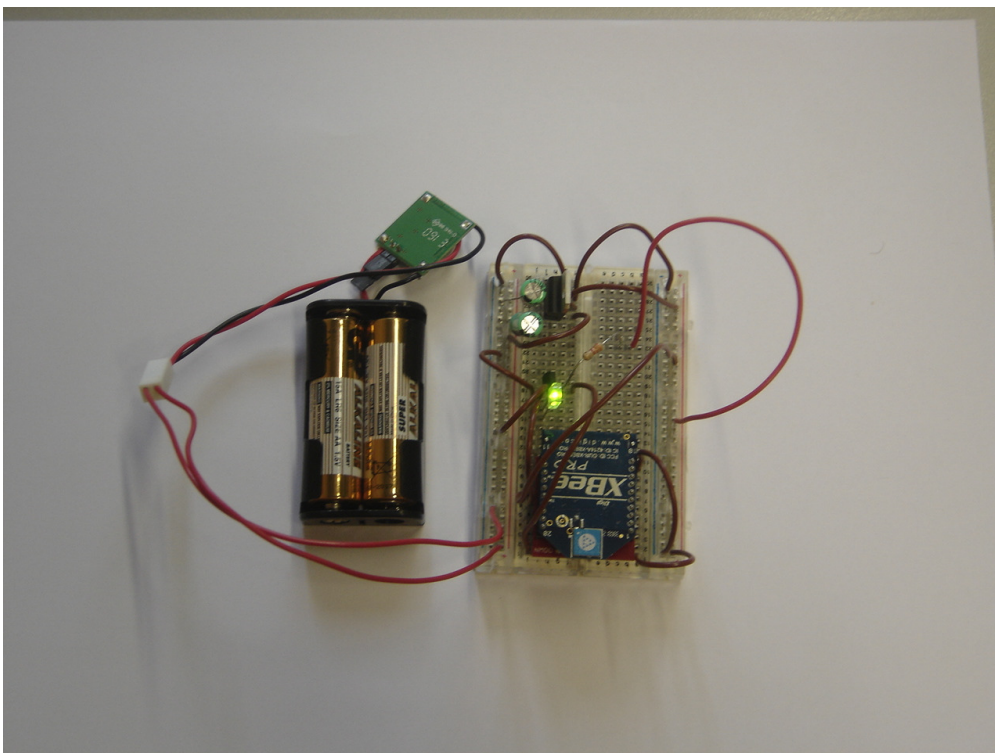


**Figura 6.4.** Diagrama de radiación de la antena whip-monopolo del módulo Xbee-Pro. Figura obtenida de [11].



**Figura 6.5.** Diagrama de radiación de la antena chip del módulo Xbee-Pro. Figura obtenida de [11].

Para alimentar los dispositivos, se preparó un circuito con dos pilas. La figura 6.6 muestra el resultado.



**Figura 6.6.** Dispositivo XBee-Pro alimentado por baterías.



Hay que explicar que en las medidas realizadas en este trabajo con los dispositivos XBee-Pro, no aparecen potencias recibidas mayores de -40 dBm, ya que en las hojas de especificaciones indica que valores tan altos no los lee (figura 6.7). Para que los valores fueran tan altos, el receptor y el transmisor deberían estar muy cerca, prácticamente una antena al lado de la otra, y este caso no es significativo.

#### DB (Received Signal Strength) Command

<Diagnostics> DB parameter is used to read the received signal strength (in dBm) of the last RF packet received. Reported values are accurate between -40 dBm and the RF module's receiver sensitivity.

AT Command: ATDB

Parameter Range [read-only]:  
0x17-0x5C (XBee), 0x24-0x64 (XBee-PRO)

Absolute values are reported. For example: 0x58 = -88 dBm (decimal). If no packets have been received (since last reset, power cycle or sleep event), "0" will be reported.

**Figura 6.7.** Lectura de RSSI en XBee-Pro.

Por último, cabe destacar también que aunque la tasa de transmisión máxima es de 250 Kbps, los dispositivos tienen unos parámetros estándar para definir la tasa de transmisión, como se ve en la figura 6.8. Durante todo el estudio, estas serán las velocidades con las que se ha trabajado.

Parameter Range: 0 - 7 (standard rates)  
0x80-0x3D090 (non-standard rates up to 250 Kbps)

Parameter	Configuration (bps)
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600
7	115200

Default Parameter Value: 3

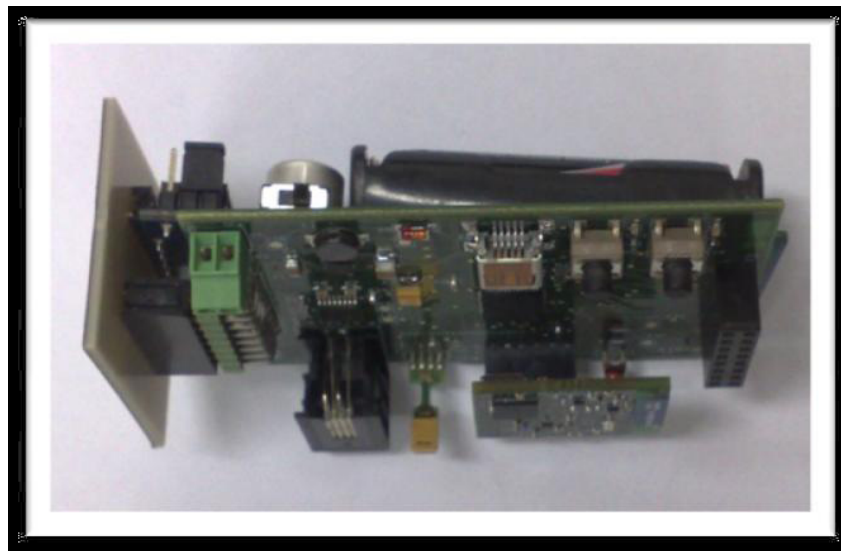
**Figura 6.8.** Velocidades de transmisión estándares para XBee-Pro.

## 6.2.- Dispositivos de la Universidad de Valencia

Los dispositivos que fueron enviados de la Universidad de Valencia para que se realizara un análisis de la parte radio, consistían en varias motas. Cada una de estas motas incluía un sensor de humedad y otro de temperatura, un procesador, puertos de expansión y alimentación, un transceptor CyFi y baterías. En la figura 6.9 se muestra un prototipo de las motas, y en la figura 6.10 se puede ver el montaje completo de una mota.



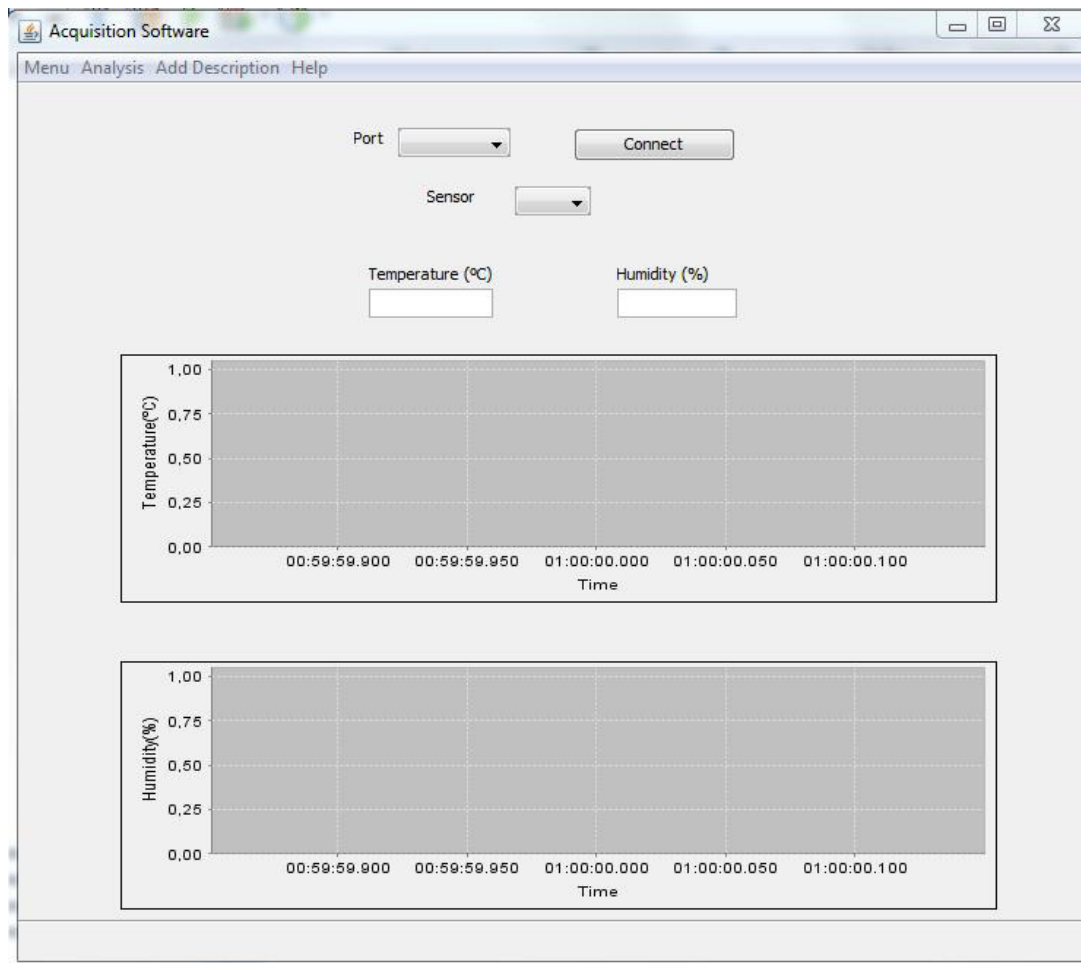
**Figura 6.9.** Prototipo de la Mota CyFi de la Universidad de Valencia.



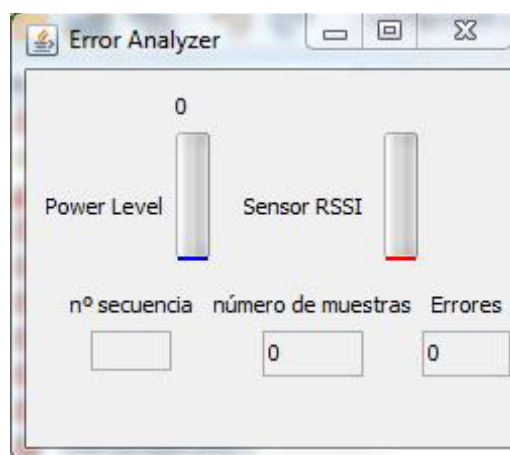
**Figura 6.10.** Mota CyFi de la Universidad de Valencia.

También se adjuntaba un software realizado en Java para la adquisición, monitorización y almacenamiento de los datos obtenidos por las motas. En las siguientes figuras se muestran dos capturas de pantalla del software:





**Figura 6.11.** Software para los sensores de las motas CyFi.



**Figura 6.12.** Software para el RSSI y nivel de potencia transmitida de las motas CyFi.

La topología básica de estos dispositivos es en estrella, en donde un nodo maestro administra un número concreto de nodos periféricos en modo esclavo. Los nodos esclavos envían periódicamente los datos obtenidos por los sensores de humedad y temperatura. Cuando no tienen que transmitir, los nodos permanecen en estado de bajo consumo. El nodo maestro se conecta a un PC mediante un cable USB.



Debido a que con las motas no se adjuntaba ningún tipo de información sobre la parte radio, hubo que obtener la información directamente de las páginas web de los fabricantes. A continuación se muestra la información más relevante al respecto.

El módulo radio era el modelo AWA24S de artaflex. Es un modelo que trabaja a 2.4 GHz, usa DSSS y que incluye el transceiver radio de Cypress CyFi CYRF7936 con un amplificador de potencia integrado. Sus características más importantes para este trabajo son las siguientes (figura 6.13):

- Opera en la banda ISM de 2.4 GHz.
- Potencia de transmisión máxima = 4 dBm.
- Sensibilidad máxima de hasta -97 dBm.
- Tamaño = 25.4 mm x 20.3 mm.
- Peso = 3 gr.
- Rango en escenarios exteriores = 1 km.
- Tasa máxima de transmisión = 250 Kbps.
- Consumo de corriente (transmitiendo -5 dBm) = 21 mA.
- Consumo de corriente (sleep mode) = 15  $\mu$ A.

Parameter Description	Conditions	Min	Typ	Max	Unit
RF frequency range	Refer Note 19	2.400	-	2.497	GHz
<b>Receiver</b> (T = 25 °C, V <sub>CC</sub> = 3.0 V, f <sub>OSC</sub> = 12.000000 MHz, BER < 1E-3)					
Sensitivity 125 kbps 64-8DR	BER 1E-3		-97	-	dBm
Sensitivity 250 kbps 32-8DR	BER 1E-3		-93	-	dBm
Sensitivity	CER 1E-3	-80	-87	-	dBm
Sensitivity GFSK	BER 1E-3, ALL SLOW = 1		-84	-	dBm
LNA gain		-	22.8	-	dB
ATT gain		-	-31.7	-	dB
Maximum received signal	LNA On	-15	-6	-	dBm
RSSI value for PWR <sub>m</sub> -60 dBm	LNA On		21	-	Count
RSSI slope			1.9	-	dB/Count
<b>Interference Performance</b> (CER 1E-3)					
Co-channel Interference rejection carrier-to-Interference (C/I)	C = -60 dBm	-	9	-	dB
Adjacent ( $\pm 1$ MHz) channel selectivity C/I 1 MHz	C = -60 dBm	-	3	-	dB
Adjacent ( $\pm 2$ MHz) channel selectivity C/I 2 MHz	C = -60 dBm	-	-30	-	dB
Adjacent ( $\geq 3$ MHz) channel selectivity C/I $\geq 3$ MHz	C = -67 dBm	-	-38	-	dB
Out-of-band blocking 30 MHz-12.75 MHz <sup>[20]</sup>	C = -67 dBm	-	-30	-	dBm
Intermodulation	C = -64 dBm, $\Delta f = 5,10$ MHz	-	-36	-	dBm
<b>Receive Spurious Emission</b>					
800 MHz	100 kHz ResBW	-	-79	-	dBm
1.6 GHz	100 kHz ResBW	-	-71	-	dBm
3.2 GHz	100 kHz ResBW	-	-65	-	dBm
<b>Transmitter</b> (T = 25°C, V <sub>CC</sub> = 3.0V)					
Maximum RF transmit power	PA = 7	+2	4	+6	dBm
Maximum RF transmit power	PA = 6	-2	0	+2	dBm
Maximum RF transmit power	PA = 5	-7	-5	-3	dBm
Maximum RF transmit power	PA = 0	-	-35	-	dBm
RF power control range		-	39	-	dB
RF power range control step size	Seven steps, monotonic	-	5.6	-	dB
Frequency deviation min	PN code pattern 10101010	-	270	-	kHz
Frequency deviation max	PN code pattern 11110000	-	323	-	kHz
Error vector magnitude (FSK error)	>0 dBm	-	10	-	%rms
Occupied bandwidth	-6 dBc, 100 kHz ResBW	500	876	-	kHz

Figura 6.13. Características del transceiver CyFi CYRF7936 de Cypress. Figura obtenida de [13].

Por último, se van a mostrar las características de la antena que usan las motas. La antena es el modelo Rufa 2.4 GHz SMD A5839 de Antenova. Trabaja en el rango 2.4-2.5 GHz y su ganancia media es de -1.2 dBi, siendo por lo tanto una antena de prestaciones moderadas. En la figura 6.15 se muestran los diagramas de radiación.



Figura 6.14. Antena usada por las motas.

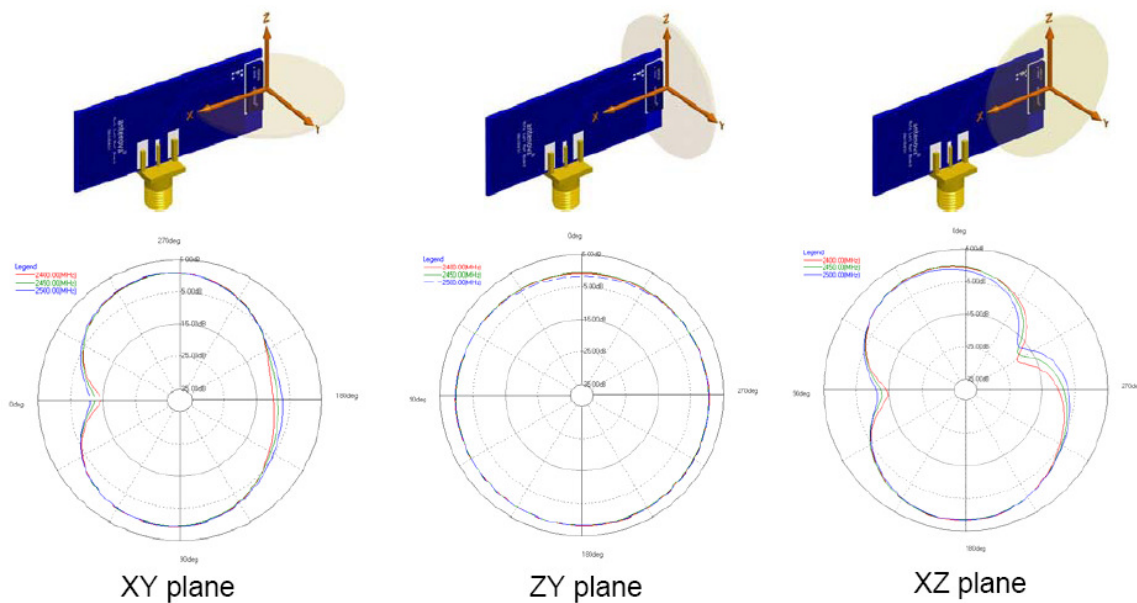


Figura 6.15. Diagramas de radiación para la antena de las motas. Figura obtenida de [14].



## 7.-Resultados obtenidos

En este apartado se describen los procedimientos seguidos para hacer las medidas en cada uno de los escenarios analizados. También se muestran los datos obtenidos en dichas medidas.

Para realizar el estudio, se han realizado tres tipos de medidas dependiendo del escenario y el objetivo en cada caso:

Por un lado, la medida de la potencia recibida (RSSI), donde se obtendrá el valor de la potencia recibida en dBm en los puntos más significativos de los escenarios. Con esta medida, y teniendo en cuenta la sensibilidad de los nodos, se podrá valorar cuál es la mejor topología y posicionamiento de los nodos (tanto transmisor como receptor) para optimizar el funcionamiento de la red.

En segundo lugar, se realizarán medidas del PER (Packet Error Rate). Este parámetro es el que se utilizará como indicador de la calidad en la transmisión.

Por último, el consumo de los nodos. Como ya se ha visto en apartados anteriores, el consumo de los nodos en una red de sensores inalámbricos es un parámetro muy importante, por lo que los datos obtenidos serán muy valiosos.

### 7.1.-Métodos para la obtención de los datos

Durante la obtención de los datos en los diferentes escenarios, los métodos utilizados han ido cambiando y evolucionando dependiendo de las necesidades y objetivos en cada caso. A continuación se explican estos métodos.

#### 7.1.1.-Potencia recibida (RSSI)

Para medir la potencia recibida se han utilizado dos métodos diferentes:

##### 7.1.1.1-Analizadores de espectro

Mediante analizadores de espectros, observando el espectro en cada punto del espacio, se puede obtener el valor de la potencia recibida en dBm.

En los casos en los que hay uno o más transmisores y se requiere hacer un barrido de todo el escenario para obtener la potencia que llega a cada zona del

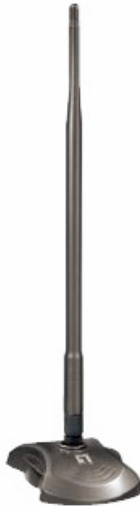
escenario, el analizador de espectros utilizado ha sido el Agilent FieldFox N9912A [9] (Figura 7.1).



**Figura 7.1.** Analizador de espectros Agilent FieldFox N9912A.

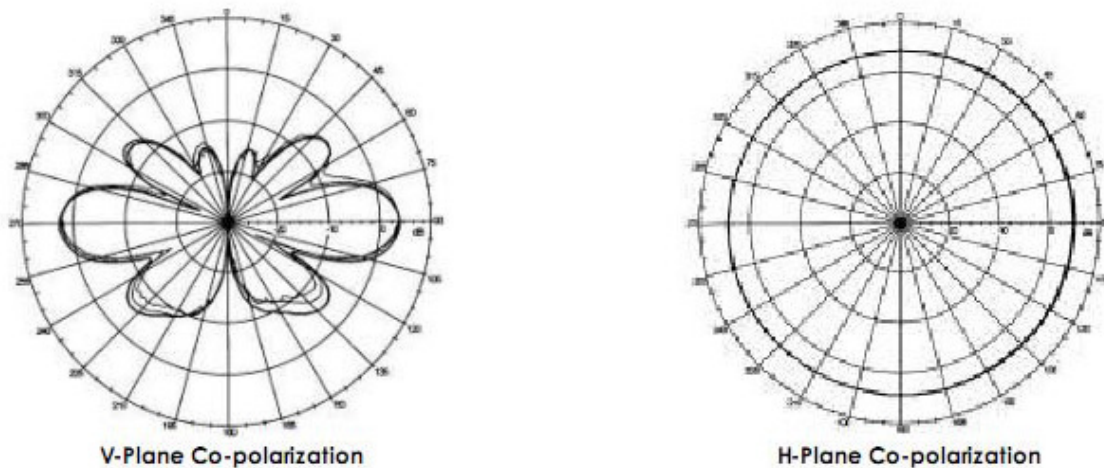
Es un analizador portátil, de reducido tamaño, ya que para obtener datos en multitud de posiciones sería inviable desplazar un analizador de espectros de gran tamaño y peso.

La antena utilizada fue el modelo OAN-1070 de Level One. Es una antena dipolo diseñada especialmente para colocarla en interiores y trabajar a una frecuencia de 2.4 GHz. El alcance máximo especificado por el fabricante es de 450 metros, por lo que es suficiente para el propósito de este trabajo. La figura 7.2 muestra una imagen de la antena, y en la figura 7.3, los diagramas de radiación.



**Figura 7.2.** Antena OAN-1070 de Level One. Figura obtenida de [15].

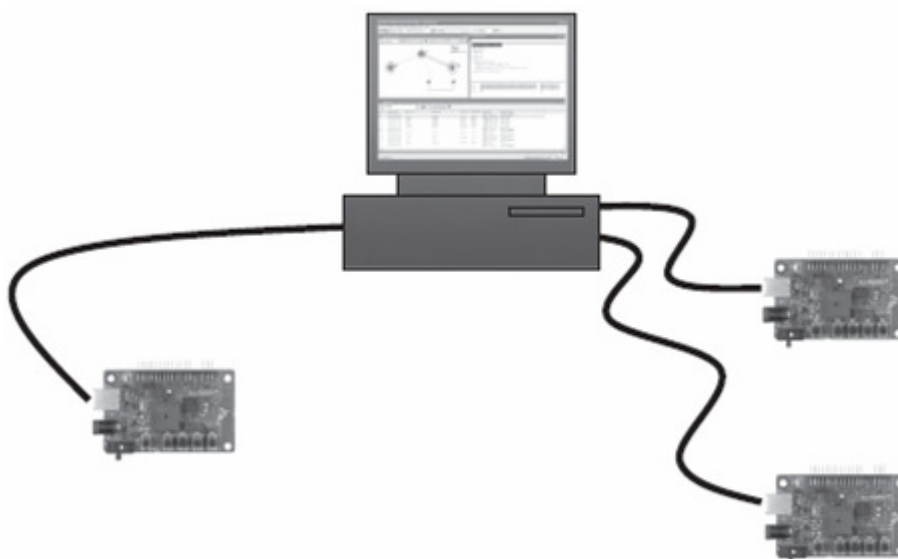
Debido a las características de radiación, para lograr unos valores lo más cercanos a la realidad posibles, la antena se colocó de manera que el máximo de transmisión alcanzara la antena en su máximo valor de directividad, en este caso, en medio de la antena.



**Figura 7.3.** Diagramas de radiación de la antena OAN-1070 de Level One. Figura obtenida de [15].

#### 7.1.1.2-Dispositivos ZigBee

En la mayoría de los casos, la manera de obtener los datos relativos a la potencia recibida se ha realizado directamente con los propios dispositivos inalámbricos. Para la configuración y cambio de parámetros, son necesarias las herramientas y aplicaciones informáticas que adjuntan las empresas con los dispositivos físicos. Estos parámetros son configurados mediante un PC, que estará conectado mediante USB a los dispositivos. En la figura 7.4 se observa un esquema del montaje necesario para configurar los dispositivos y leer los datos necesarios.



**Figura 7.4.** Montaje PC + dispositivos ZigBee. Figura obtenida de [1].

Por comodidad a la hora de leer los datos que interesan y que están almacenados en los dispositivos, se creó un programa en Java, llamado 'ZigBeePERReceiver', más cómodo y de más fácil acceso a los datos. El código del programa puede consultarse en el anexo B. En las figuras 7.5 y 7.6 se muestran capturas de pantalla del programa utilizado para obtener el dato de la potencia recibida por el dispositivo.

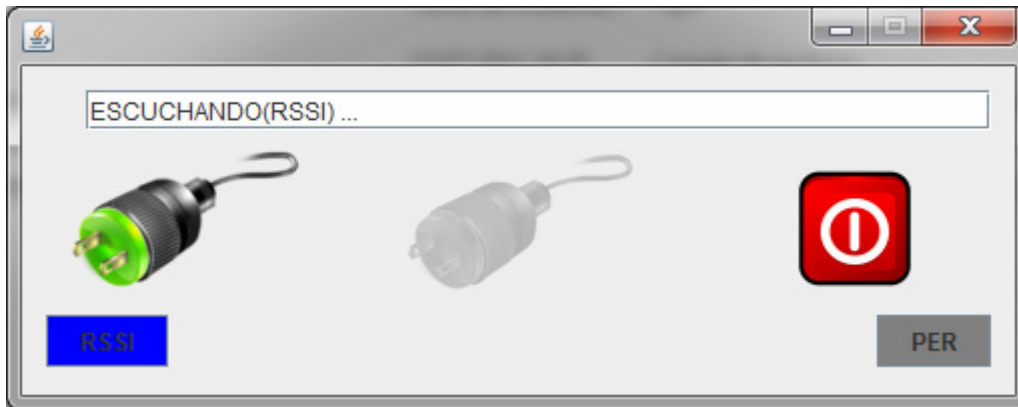


Figura 7.5. Programa en Java para la lectura de RSSI.

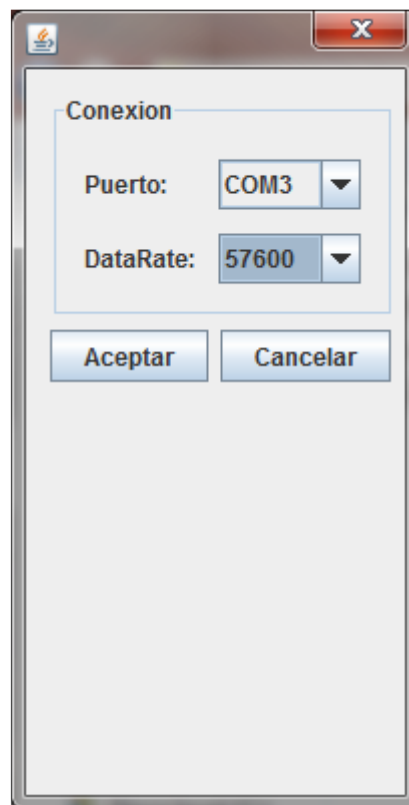


Figura 7.6. Elección del puerto USB y velocidad de transmisión.

Por las mismas razones, también se creó un programa para definir las características del transmisor. Este programa se llamó 'ZigBeePERSender', y también se puede consultar su código en el anexo B. En las figuras 7.7 y 7.8 se

muestran dos capturas de pantalla de dicho programa. Notar que en la figura 7.7 se observa la posibilidad de realizar los envíos con ACK o sin ACK.



Figura 7.7. Programa para el envío de RSSI.

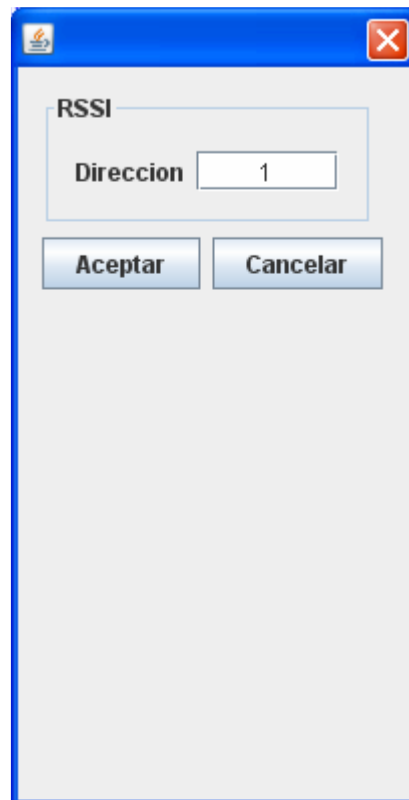


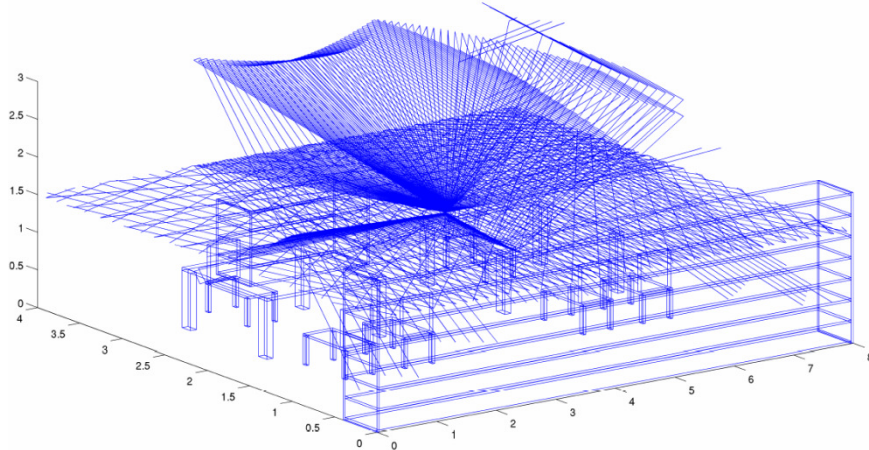
Figura 7.8. Elección de dirección de destino.

### 7.1.1.3-Simulación con trazado de rayos

Una tercera forma de obtener datos sobre la potencia recibida, ha sido mediante la simulación de algunos escenarios con un software de trazado de rayos en tres dimensiones. El programa utilizado fue desarrollado completamente en la Universidad Pública de Navarra con la herramienta Matlab.

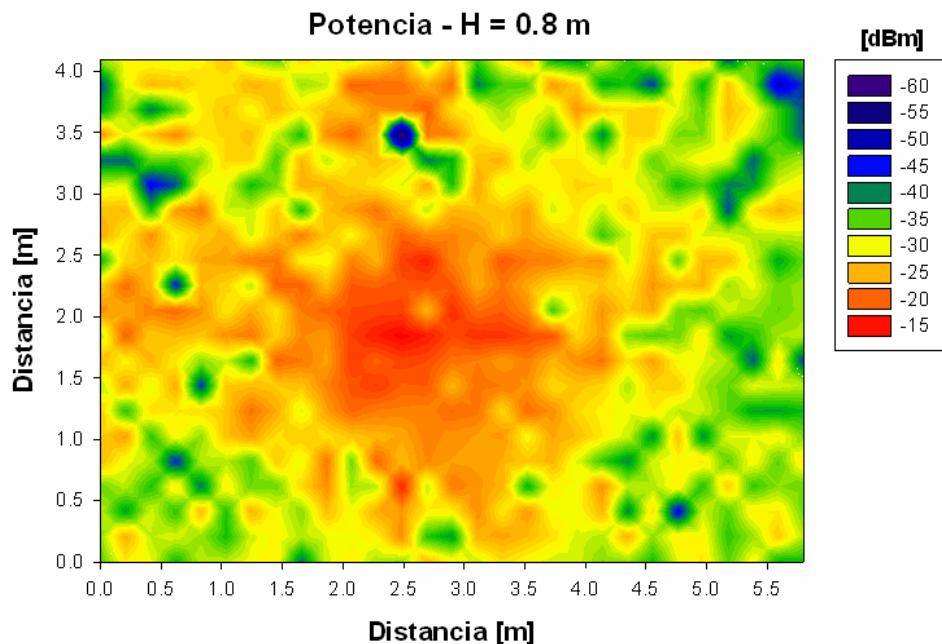


Como se ha mencionado anteriormente en este trabajo, el método de trazado de rayos que utiliza este software se basa en un algoritmo de *Ray-Launching*. En la figura 7.9 se puede ver un ejemplo.



**Figura 7.9.** Escenario interior en 3D con los rayos lanzados desde un transmisor situado en el centro.

Cada rayo se computa individualmente, y el campo eléctrico de cada punto del escenario es la suma de todas las contribuciones de los rayos que los atraviesan. Al calcular la potencia recibida en cada punto del espacio, pueden crearse mapas de cobertura a una altura fija, como se muestra en la figura 7.10.

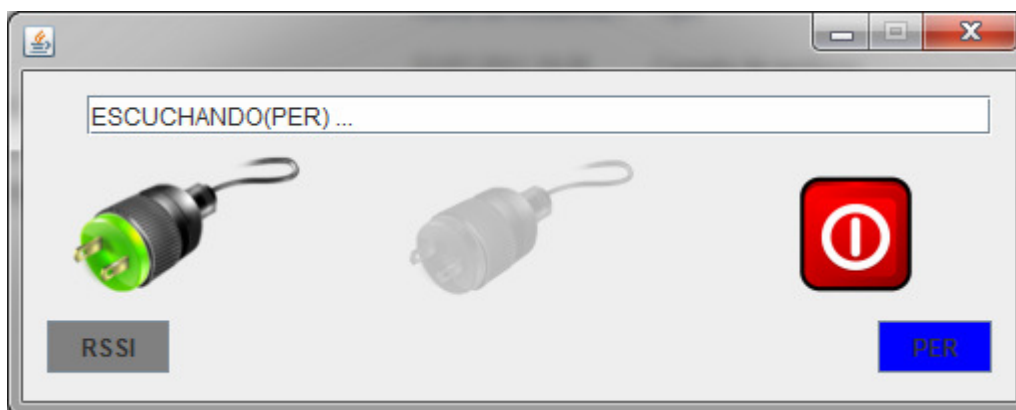


**Figura 7.10.** Mapa de potencia recibida mediante trazado de rayos 3D, a una altura de 0.8 metros, con un único transmisor situado en el centro.

### 7.1.2.-Calidad (PER)

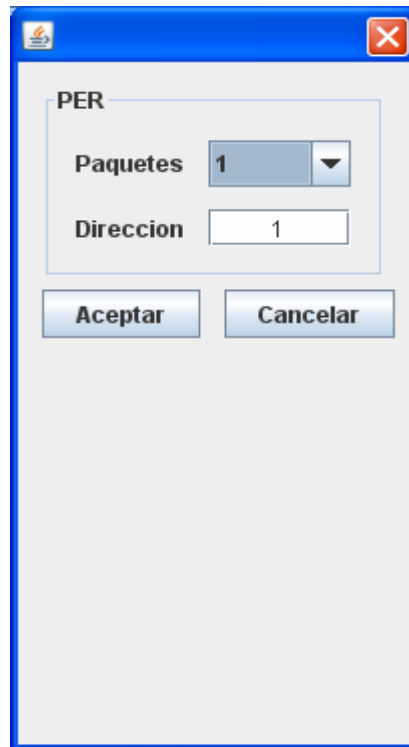
El parámetro que he utilizado para medir y comparar la calidad de los enlaces ha sido el PER (Packet Error Rate). En principio la idea era utilizar un sniffer inalámbrico que proporcionara el número de paquetes perdidos, pero debido a la dificultad de adquirir uno (principalmente el precio), y varios problemas que hubo con uno que sí se pudo adquirir, el método elegido para obtener estos datos fue el mismo que para RSSI. Es decir, se utilizó el valor del PER proporcionado por los propios dispositivos inalámbricos. En el anexo C se pueden ver los datos y los problemas que hubo con el sniffer que se adquirió.

Para ello, el mismo programa creado para leer los valores de RSSI tiene la opción de leer el PER. En la figura 7.11 se ve la opción para el PER activada (en azul).



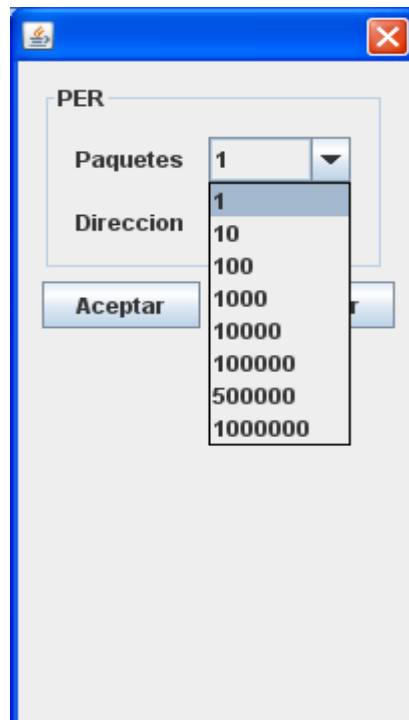
**Figura 7.11.** Programa en Java para la lectura del PER.

Para el transmisor, también se utiliza el mismo programa que para el RSSI, ya que dispone de una opción para analizar el PER (figura 7.7). Pero a diferencia del envío de paquetes para medir la potencia recibida, en el caso de activar la opción de PER, aparece una nueva ventana para ajustar diferentes parámetros. En la figura 7.12 se muestra esta nueva ventana. Los parámetros que se pueden introducir son la dirección de destino de los paquetes y el número de paquetes a transmitir.



**Figura 7.12.** Programa en Java para el envío de paquetes en el caso del PER.

El número de paquetes a transmitir sólo puede ser uno de los valores que aparecen en la lista desplegable, aunque esta lista puede ser modificada si es necesario. En la figura 7.13 se observan los valores predeterminados de la lista desplegable con los que se trabajó habitualmente.



**Figura 7.13.** Valores de número de paquetes a transmitir el caso del PER.

### 7.1.3.-Consumo

Para obtener datos sobre el consumo de los dispositivos se utilizó un osciloscopio modelo Tektronix DPO 3014 como el mostrado en la figura 7.14.



**Figura 7.14.** Osciloscopio Tektronix DPO 3014.

Para lograr los datos de consumo de corriente, se introdujo una resistencia de  $1 \Omega$  en paralelo en el circuito de alimentación de los dispositivos. Así, midiendo con el osciloscopio la diferencia de voltaje en dicha resistencia, se logra un valor aproximado del valor de la corriente. En la figura 7.15 se muestra la fórmula de la ley de Ohm, donde si sustituimos  $R = 1$ , nos queda que la corriente es del mismo valor que el voltaje.

$$I = \frac{V}{R}$$

**Figura 7.15.** Ley de Ohm.

Al trabajar con los dispositivos XBee-Pro, el nivel de potencia al que transmiten es un valor fijado por el usuario. Este valor no puede ser variado automáticamente por el dispositivo, por lo que el consumo de corriente dependerá del nivel especificado por el usuario, y se mantendrá aproximadamente constante.

## 7.2.-Resultados obtenidos

En este apartado se describirán los escenarios y medidas realizadas en cada uno de ellos.

### 7.2.1.-Escenario 1: RSSI (Received Signal Strength Indicator)

En este escenario se realizaron medidas de la potencia recibida o RSSI. Concretamente, se midió la distribución de potencia dentro de un escenario interior, situando un único nodo transmisor. Para medir la potencia que se recibe en cada punto, se ha utilizado el analizador de espectros Agilent FieldFox N9912A junto con la antena OAN-1070 de Level One.

Los dispositivos usados son los XBee-Pro, transmitiendo a su máxima potencia, es decir 18 dBm.

El escenario está situado en el edificio Jerónimo de Ayanz de la UPNA. Se trata del laboratorio L5 y el despacho D5. En la figura 7.16 y en la 7.17 se muestran los planos del laboratorio y del despacho donde se realizaron las medidas. El despacho está contiguo al laboratorio, y están comunicados por una puerta.

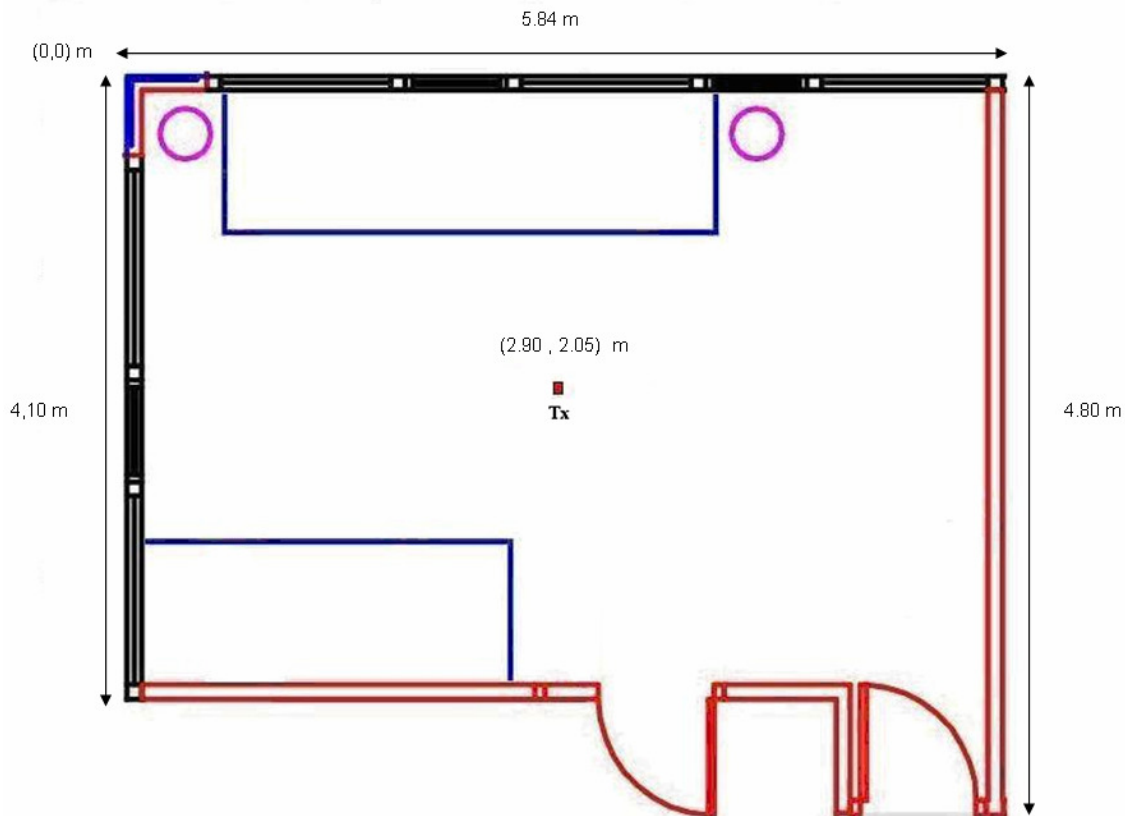
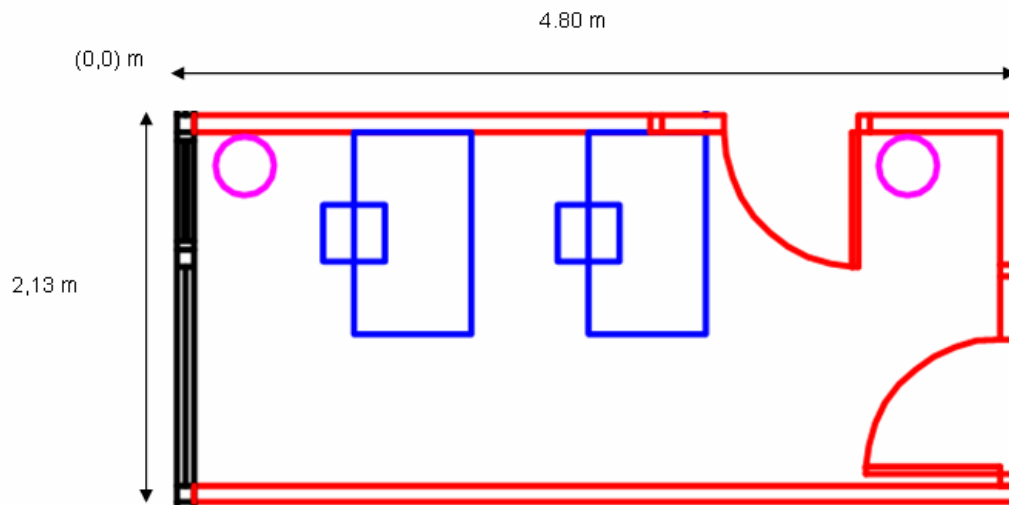


Figura 7.16. Plano del laboratorio L5.



**Figura 7.17.** Plano del despacho D5.

Ambos espacios presentan características muy similares. Aunque difieren en el tamaño, los dos contienen mesas, cajones, columnas de hormigón, varias ventanas de cristal y paredes tanto de pladur como de hormigón. Por lo que se puede decir que ambos son escenarios interiores adecuados para el estudio que se desean realizar.



**Figura 7.18.** Fotografía del laboratorio L5.

Para indicar las posiciones, se han usado coordenadas cartesianas ( $x =$  horizontal,  $y =$  vertical). El centro de coordenadas está situado en la esquina superior izquierda de cada plano.



El nodo transmisor se colocó en el centro del laboratorio, marcado en rojo en la figura 7.16, en las coordenadas (2.90, 2.05). La antena del transmisor se situó a 1.05 metros de altura, y debido al diagrama de radiación de la antena del analizador de espectros, se colocó a la altura de 1.05 metros el centro de esta antena, para recibir el máximo. En la tabla 7.1 y la 7.2 se muestran los valores obtenidos en las medidas. Las unidades de la potencia recibida son dBm y las coordenadas son metros.

(x,y)	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5
0.5	-40.33	-36.37	-32.23	-36.35	-35.34	-36.27	-40.05	-48.74	-39.52	-39.99	-42.98
1	-35.73	-31.76	-32.56	-29.36	-31.62	-31.65	-38.59	-42.93	-44.15	-35.99	-45.96
1.5	-43.17	-37.51	-27.42	-26.16	-26.53	-29.71	-43.64	-44.59	-39.11	-49.49	-42.31
2	-36.18	-33.53	-36.03	-29.77	-23.07	x	-41.81	-44.76	-46.96	-46.09	-45.13
2.5	-44.60	-51.07	-43.36	-39.02	-45.41	-22.37	-25.75	-31.40	-37.85	-39.92	-40.79
3	-44.48	-37.61	-40.27	-42.92	-40.10	-26.71	-29.58	-30.51	-32.54	-49.87	-42.08
3.5	x	-53.36	-49.66	-42.77	-31.69	-32.18	-29.86	-34.82	-32.89	-34.18	-43.35
4	x	x	x	x	x	x	x	x	x	-50.29	-36.95
4.5	x	x	x	x	x	x	x	x	x	-37.40	-41.55

**Tabla 7.1.** Distribución de potencia en el laboratorio L5.

(x,y)	1,2	2,4	3,6
1	-46.11	-43.72	-55.46
2	-51.76	-50.56	-42.86

**Tabla 7.2.** Distribución de potencia en el despacho D5.

Después de mostrar los datos obtenidos empíricamente, a continuación se muestran los valores obtenidos teóricamente, utilizando diferentes modelos de propagación. En la figura 7.19 se puede ver una captura de pantalla del documento excel utilizado para el cálculo con los diferentes modelos. Hay que destacar que varios de estos modelos fueron desarrollados para frecuencias de hasta 1.800 MHz, lo que resulta una pequeña limitación en frecuencia, ya que los cálculos realizados fueron para 2.450 MHz.



	A	B	C	D	E	F
1	Frecuencia de Tx.	2450,00	MHz			
2	Potencia de Tx.	18,00	dBm's			
3	Ganancia Antena Tx.	0,00	dB'i's			
4	Ganancia Antena Rx.	0,00	dB'i's			
5	Altura de Antena Tx.	1,05	mts.			
6	Altura de Antena Rx.	1,05	mts.			
7	Coordenadas (x,y,x0,y0)	3,6	5,1	2,9	2,05	
8	<b>METROS</b>	<b>3,129297046</b>				
9	Modelo de Propagación de Espacio Libre	50,13				
10	Modelo de Propagación COST-231, interior = MultParedes.	50,13				
11	Modelo de Propagación ITU-R P.1238	53,66				
12	Modelo de Propagación Linear Path Atenuation Model	51,60				
13	Modelo de Propagación Keenan-Motley	50,13				
14	Modelo de Propagación Multi-Wall	50,13				
15		-32,13				
16		-32,13				
17	Potencia Recibida	-35,66				
18		-33,60				
19		-32,13				
20		-32,13				
21						
22						
23						
24	<b>PERDIDAS DE ESPACIO LIBRE REFERENCIA A 1MT.</b>	<b>40,22332169</b>				
25						
26						
27						

Figura 7.19. Cálculo con modelos de propagación.

Dentro del laboratorio, los valores teóricos para los modelos de propagación Cost-231, Keenan-Motley y Multi-Wall son iguales. Estos datos se muestran en la tabla 7.3.

(x,y)	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5
0.5	-31,34	-30,01	-28,62	-27,29	-26,31	-26,05	-26,64	-27,80	-29,18	-30,56	-31,84
1	-30,59	-28,96	-27,08	-25,04	-23,24	-22,69	-23,87	-25,86	-27,86	-29,64	-31,18
1.5	-30,05	-28,15	-25,77	-22,69	-18,87	-17,17	-20,44	-24,02	-26,79	-28,96	-30,71
2	-29,83	-27,80	-25,15	-21,32	-14,33	-3,19	-17,82	-23,06	-26,31	-28,67	-30,52
2.5	-29,98	-28,04	-25,57	-22,28	-17,82	-15,50	-19,72	-23,72	-26,64	-28,86	-30,65
3	-30,46	-28,77	-26,79	-24,56	-22,49	-21,83	-23,24	-25,47	-27,62	-29,48	-31,07
3.5	-31,18	-29,79	-28,31	-26,87	-25,77	-25,47	-26,14	-27,42	-28,91	-30,36	-31,70
4	x	x	x	x	x	x	x	x	x	-31,37	-32,46
4.5	x	x	x	x	x	x	x	x	x	-32,40	-33,28

Tabla 7.3. Valores para Cost-231, Keenan-Motley y Multi-Wall para L5.

Los demás modelos dan unos resultados sensiblemente diferentes a los anteriores. En la tabla 7.4 se muestran los datos para el modelo ITU-R P.1238, y en la tabla 7.5 los resultados para el modelo Linear Path Atenuation.





(x,y)	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5
0.5	-34,55	-32,69	-30,74	-28,88	-27,50	-27,14	-27,96	-29,59	-31,52	-33,45	-35,25
1	-33,49	-31,21	-28,59	-25,73	-23,20	-22,43	-24,09	-26,88	-29,68	-32,16	-34,32
1.5	-32,74	-30,08	-26,75	-22,43	-17,09	-14,71	-19,28	-24,30	-28,18	-31,21	-33,67
2	-32,43	-29,59	-25,88	-20,52	-10,74	4,86	-15,61	-22,95	-27,50	-30,81	-33,40
2.5	-32,64	-29,92	-26,47	-21,86	-15,61	-12,37	-18,29	-23,88	-27,96	-31,08	-33,58
3	-33,31	-30,95	-28,18	-25,05	-22,15	-21,23	-23,20	-26,33	-29,33	-31,94	-34,16
3.5	-34,32	-32,38	-30,31	-28,28	-26,75	-26,33	-27,26	-29,07	-31,14	-33,18	-35,05
4	x	x	x	x	x	x	x	x	x	-34,59	-36,12
4.5	x	x	x	x	x	x	x	x	x	-36,03	-37,27

**Tabla 7.4.** Valores del modelo ITU-R P.1238 para el laboratorio L5.

(x,y)	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5
0.5	-32,68	-31,17	-29,60	-28,13	-27,06	-26,78	-27,42	-28,69	-30,23	-31,78	-33,27
1	-31,82	-29,98	-27,91	-25,69	-23,76	-23,18	-24,44	-26,58	-28,76	-30,74	-32,50
1.5	-31,21	-29,08	-26,48	-23,18	-19,19	-17,43	-20,82	-24,60	-27,59	-29,98	-31,96
2	-30,96	-28,69	-25,81	-21,75	-14,52	-3,24	-18,10	-23,58	-27,06	-29,66	-31,75
2.5	-31,13	-28,95	-26,47	-22,75	-18,10	-15,71	-20,08	-24,28	-27,42	-29,87	-31,89
3	-31,67	-29,77	-27,59	-25,17	-22,97	-22,27	-23,76	-26,15	-28,49	-30,56	-32,37
3.5	-32,50	-30,91	-29,26	-27,67	-26,48	-26,15	-26,87	-28,28	-29,92	-31,56	-33,10
4	x	x	x	x	x	x	x	x	x	-32,71	-33,99
4.5	x	x	x	x	x	x	x	x	x	-33,92	-34,96

**Tabla 7.5.** Linear Path Attenuation Model para el laboratorio L5.

Para los resultados teóricos para la zona del despacho D5, se ha tenido en cuenta que hay una pared en medio y que para cada uno de los modelos de propagación se estiman unas pérdidas diferentes.

Para el modelo Cost-231, se han estimado unas pérdidas adicionales de 3.4 dB debidas a que la señal atraviesa una pared fina.

(x,y)	1,2	2,4	3,6
1	-36,48	-35,42	-35,53
2	-38,48	-37,84	-37,9

**Tabla 7.6.** Modelo Cost-231 para el despacho D5.



Para los modelos Keenan-Motley y Multi-Wall, a los valores obtenidos se han añadido unas pérdidas debidas a la pared de 2 y 1 dB respectivamente.

(x,y)	1,2	2,4	3,6
1	-35,08	-34,02	-34,13
2	-37,08	-36,44	-36,50

**Tabla 7.7.** Modelo Keenan-Motley para el despacho D5.

(x,y)	1,2	2,4	3,6
1	-34,08	-33,02	-33,13
2	-36,08	-35,44	-35,50

**Tabla 7.8.** Modelo Multi-Wall para el despacho D5.

Por último, en las tablas 7.9 y 7.10 se muestran los valores obtenidos con los modelos ITU-R P.1238 y Linear Path Attenuation. En el modelo ITU no se han añadido pérdidas adicionales por la pared, ya que el modelo tiene en cuenta únicamente el suelo, no las paredes.

(x,y)	1,2	2,4	3,6
1	-34,73	-33,48	-33,60
2	-37,14	-36,36	-36,43

**Tabla 7.9.** Linear Path Attenuation Model para el despacho D5.

(x,y)	1,2	2,4	3,6
1	-36,99	-35,50	-35,66
2	-39,78	-38,88	-38,97

**Tabla 7.10.** ITU-R P.1238 para el despacho D5.

Analizando todos los resultados, se observa que las diferencias entre los datos proporcionados por los modelos son muy pocas. En todos los puntos hay aproximadamente 2 dB de diferencia máxima.

Con respecto a los datos obtenidos en las mediciones, también se puede afirmar que las diferencias existentes con los datos de los modelos son pequeñas, exceptuando algún punto concreto en el que han sido algo mayores. Pero teniendo en cuenta que en los escenarios existen multitud de obstáculos y que también transitan personas, podemos concluir que los datos obtenidos con el analizador de espectros concuerdan con los de los modelos.



### 7.2.2.-Escenario 2: RSSI + Consumo con varios transmisores

En este escenario se utilizaron los dispositivos XBee-Pro. El lugar para las medidas, fue también el laboratorio L5 del edificio de I+D Jerónimo de Ayanz de la Universidad Pública de Navarra.

El objetivo de este escenario es obtener información sobre el comportamiento de una red con un coordinador y con varios dispositivos transmitiendo a la vez. Concretamente, se estudiaron los niveles de potencia que recibe el coordinador de cada uno de los dispositivos transmisores. Para ello, se situó un dispositivo, que sería el coordinador, en el centro del laboratorio, que se corresponde con las coordenadas (2.90, 2.05), y se realizaron medidas con uno, dos y tres dispositivos transmitiendo a la vez.

Del mismo modo, se aprovechó el mismo escenario para realizar las medidas para los 5 diferentes niveles de potencia de transmisión de los que disponen los dispositivos XBee-Pro.

Igualmente, se realizaron las primeras medidas del consumo de los dispositivos XBee-Pro, que se obtuvieron con el osciloscopio DPO3014 de Tektronix.

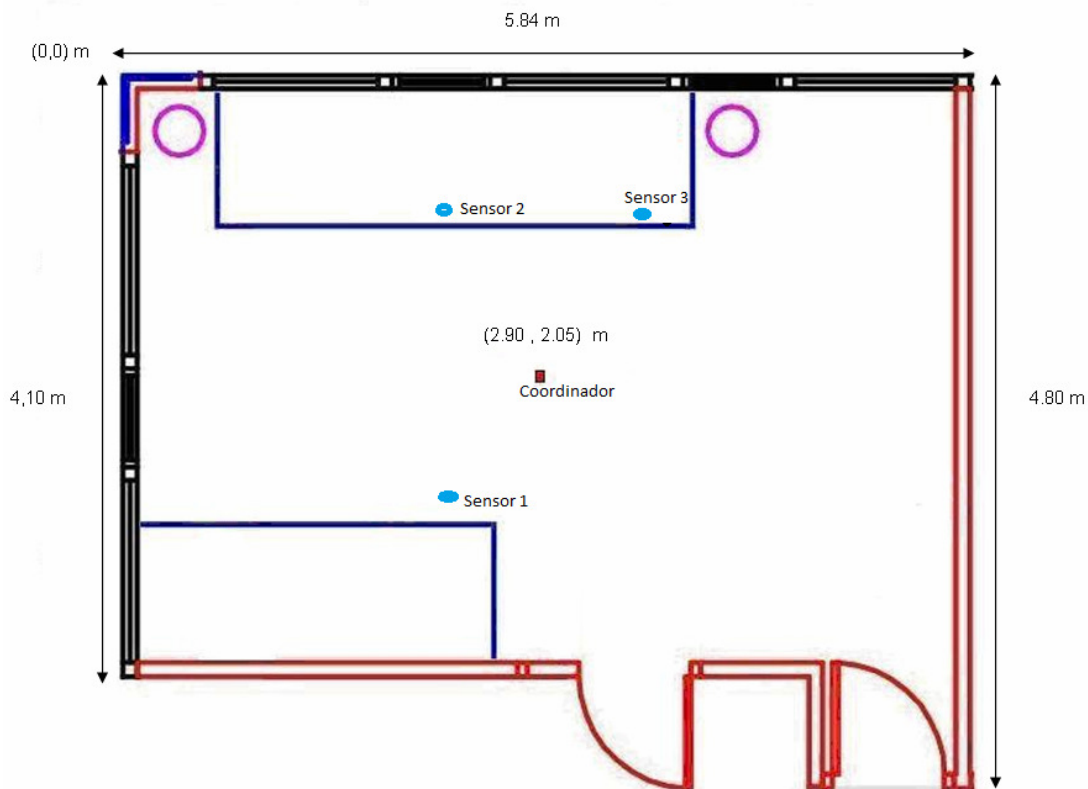
Para poder comparar los resultados mejor, los 3 transmisores se situaron a la misma distancia del coordinador (1.59m), y todos, incluido el coordinador, a una altura de 1.05 metros. Las coordenadas de los transmisores son las siguientes:

Sensor 1 = (1.80, 3.20)

Sensor 2 = (1.80, 0.90)

Sensor 3 = (4.00, 0.90)

En la siguiente figura se muestra el plano con la localización del coordinador y los 3 sensores:



**Figura 7.20.** Localización de los dispositivos para el escenario 2.

### **1- POTENCIA DE TRANSMISIÓN MÁXIMA: NIVEL 4 =18 dBm:**

El primer caso de estudio fue con la potencia de transmisión de los dispositivos a su máximo nivel, que es el nivel 4, que corresponde con 18 dBm de potencia. A continuación se muestran los resultados obtenidos para los casos con 1, 2 y 3 transmisores.

Los valores de potencia recibida de cada transmisor que aparecen a continuación se refieren al nivel de potencia que recibe el coordinador de cada uno de los transmisores. Estos valores los proporcionaba el mismo dispositivo.

#### 1.1- Un único transmisor (Sensor 1):

Potencia recibida = -53 dBm.

#### 1.2- Dos transmisores transmitiendo simultáneamente:

Potencia recibida (Sensor 1) = -50 dBm.

Potencia recibida (Sensor 2) = -51 dBm.

#### 1.3- Tres transmisores transmitiendo simultáneamente:

Potencia recibida (Sensor 1) = -53 dBm.

Potencia recibida (Sensor 2) = -48 dBm.

Potencia recibida (Sensor 3) = -38 dBm.

El consumo de los dispositivos para este nivel de potencia transmitida se observa en la figura 7.21, donde se muestra una captura de la pantalla del osciloscopio. Hay que recordar, como se explicó en el apartado 7.2, los voltios que aparecen en el osciloscopio son equivalentes a amperios.

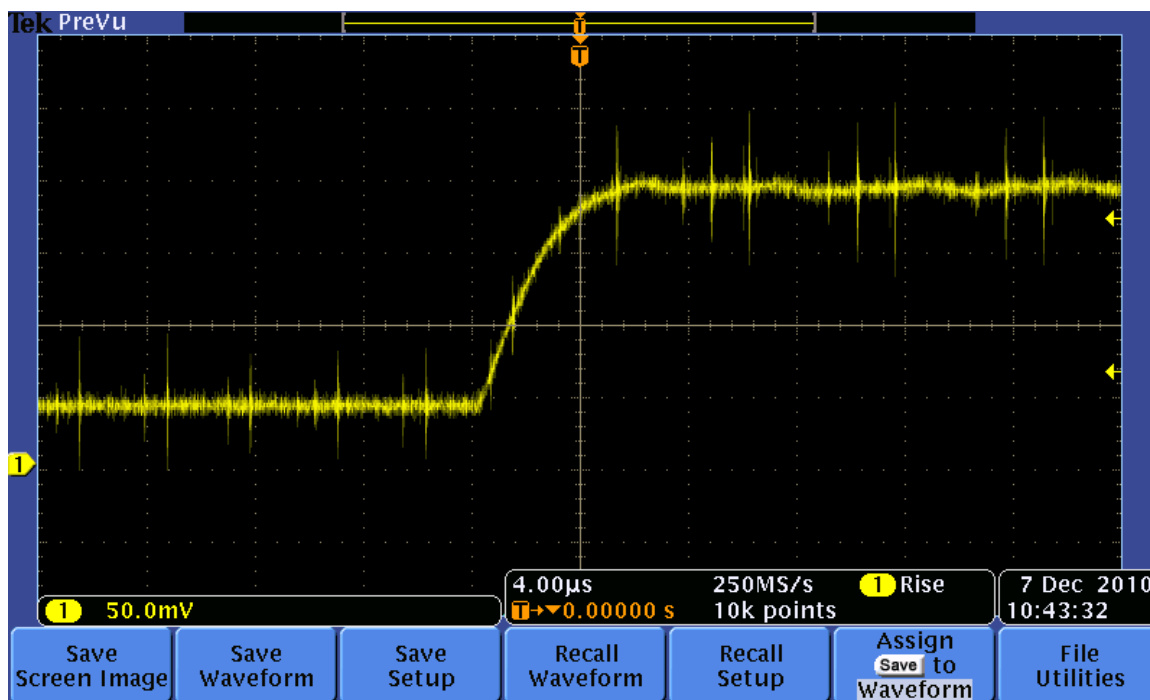


Figura 7.21. Consumo XBee-Pro a nivel 4.

El salto en la gráfica se debe al cambio de estado del sensor a modo de transmisión. Se observa que el consumo es aproximadamente 200 mA.

Con los siguientes niveles de potencia, se realizaron medidas únicamente con los 3 transmisores emitiendo simultáneamente:

## **2- POTENCIA DE TRANSMISIÓN NIVEL 3 =16 dBm:**

Potencia recibida (Sensor 1) = -54 dBm.

Potencia recibida (Sensor 2) = -42 dBm.

Potencia recibida (Sensor 3) = -31 dBm.

El consumo del dispositivo se muestra en la figura 7.22. Como se esperaba, el consumo es menor, alrededor de 125 mA.

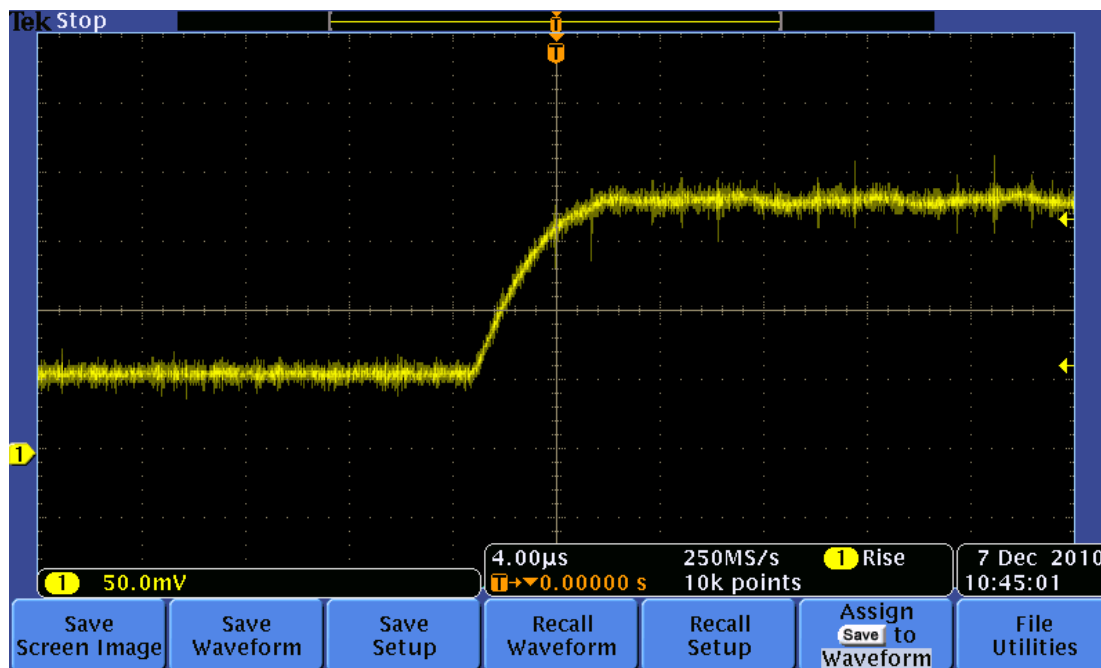


Figura 7.22. Consumo XBee-Pro a nivel 3.

### 3- POTENCIA DE TRANSMISIÓN NIVEL 2 =14 dBm:

Potencia recibida (Sensor 1) = -60 dBm.

Potencia recibida (Sensor 2) = -47 dBm.

Potencia recibida (Sensor 3) = -57 dBm.

En la figura 7.23 se observa que el consumo sigue decreciendo al decrecer el nivel de potencia transmitida. En este nivel es aproximadamente de 100 mA.

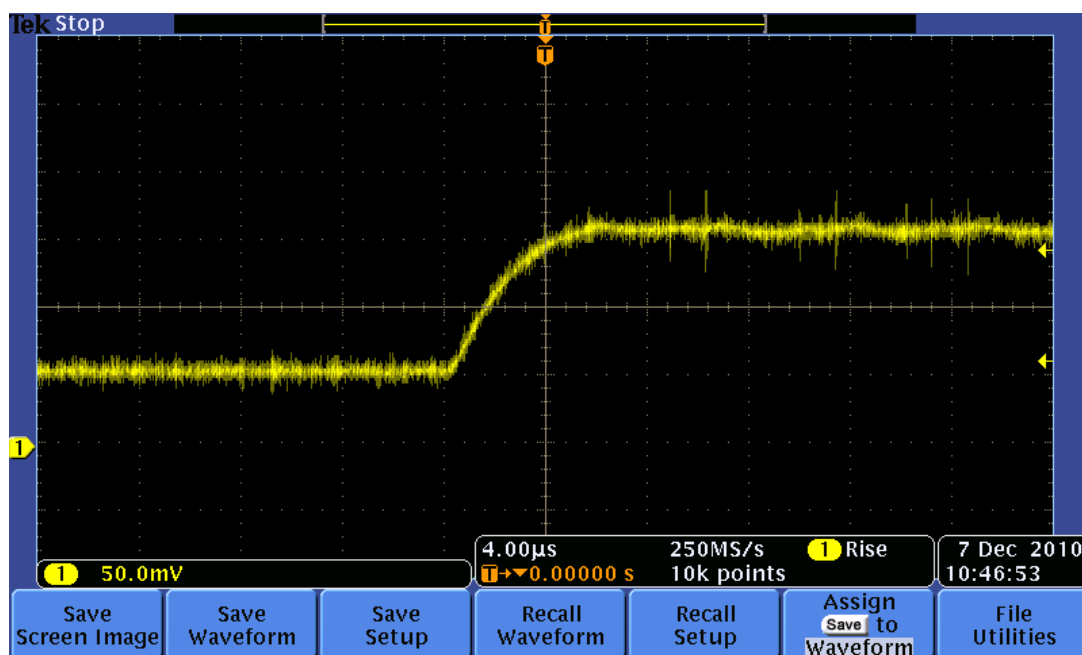


Figura 7.23. Consumo XBee-Pro a nivel 2.

#### **4- POTENCIA DE TRANSMISIÓN NIVEL 1 =12 dBm:**

Potencia recibida (Sensor 1) = -63 dBm.

Potencia recibida (Sensor 2) = -49 dBm.

Potencia recibida (Sensor 3) = -44 dBm.

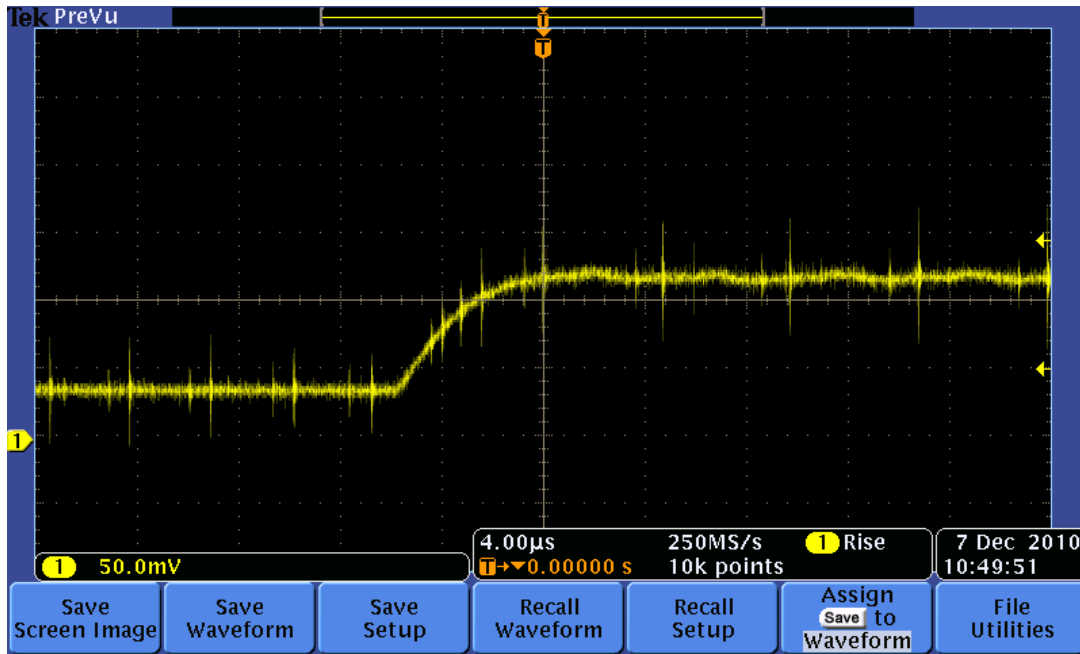


Figura 7.24. Consumo XBee-Pro a nivel 1.

El consumo ha descendido hasta aproximadamente 75 mA.

#### **5- POTENCIA DE TRANSMISIÓN NIVEL 0 =10 dBm:**

Potencia recibida (Sensor 1) = -66 dBm.

Potencia recibida (Sensor 2) = -56 dBm.

Potencia recibida (Sensor 3) = -51 dBm.

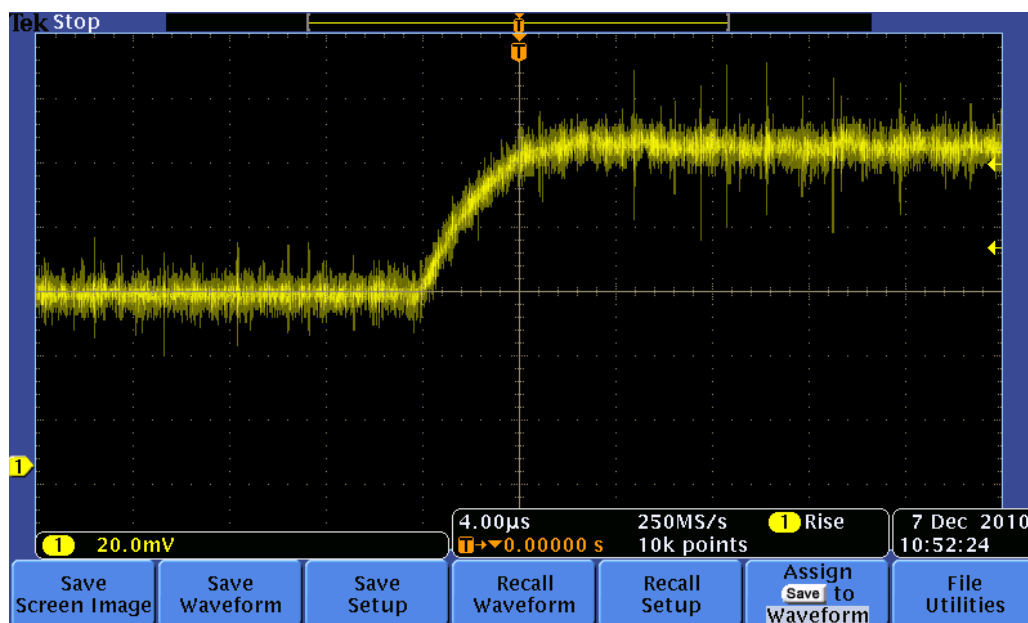


Figura 7.25. Consumo XBee-Pro a nivel 0.

En los resultados anteriores se observa claramente cómo al disminuir el nivel de potencia de transmisión de los sensores, la potencia que recibe el coordinador de cada uno de ellos disminuye igualmente, como era de esperar.

Aunque la distancia entre dispositivos es muy pequeña, se aprecia claramente que en algunos casos la caída de potencia no es exactamente la esperada. Esto es debido a los efectos descritos en el apartado de propagación en espacio libre. Además, hay que tener en cuenta las pérdidas debidas al movimiento de las personas, que afectaba significativamente a la potencia recibida, ya que durante las medidas había personas dentro del laboratorio.

También cabe destacar que aunque la distancia entre dispositivos es pequeña y que no existen obstáculos en la línea de visión directa entre los dispositivos, el entorno afecta mucho a cada transmisión, y por ello tenemos diferencias en algunos casos superiores a 10 dBm. Para comprobar que estas diferencias no son debidas a las posibles diferencias entre sensores, se realizaron medidas individualmente de los tres sensores en la misma posición, concretamente en las coordenadas (1.80, 3.20). El nivel de potencia transmitida era el 4 = 18 dBm. Los resultados obtenidos demuestran que las diferencias existentes son debidas al entorno, y no a los dispositivos transmisores:

Potencia recibida (Sensor 1) = -48 dBm.

Potencia recibida (Sensor 2) = -48 dBm.

Potencia recibida (Sensor 3) = -45 dBm.

Respecto al consumo, se aprecia claramente cómo al disminuir el nivel de potencia transmitida, el consumo también disminuye. Al máximo nivel de transmisión, que corresponde con 18 dBm, el consumo es de 200 mA. Al mínimo





nivel de transmisión, 10 dBm, el consumo desciende hasta los 45 mA aproximadamente. Por tanto, será clave conseguir que la red funcione correctamente con la mínima potencia de transmisión posible, ya que el ahorro en el consumo es muy importante. En la figura 7.26 se muestra una gráfica con los diferentes niveles de consumo.

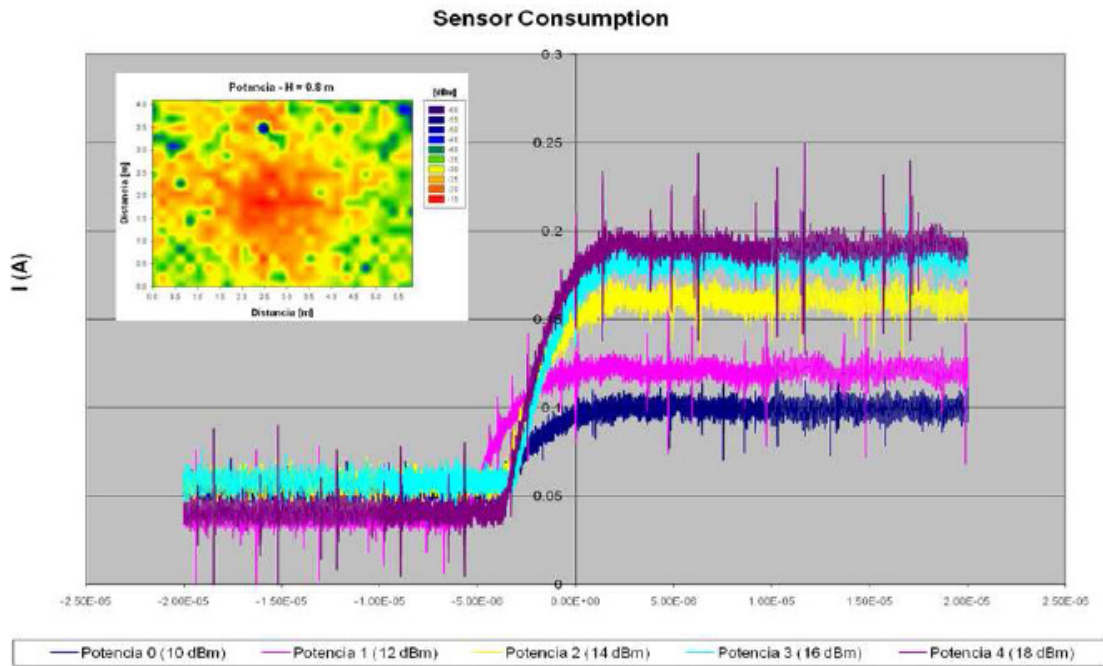


Figura 7.26. Comparación del consumo según el nivel de potencia transmitido.



### 7.2.3.-Escenario 3: RSSI con diferentes obstáculos

En este escenario se estudió cómo la potencia recibida varía dependiendo de los diferentes obstáculos que se encuentra la señal en la transmisión. Las medidas se realizaron en el mismo laboratorio de los escenarios anteriores, el L5 del edificio Jerónimo de Ayanz. En la figura 7.16 se puede observar un plano del mismo.

Para realizar las medidas, se configuró una red con un solo transmisor y un receptor. El receptor se colocó en el centro del laboratorio, cuyas coordenadas son (2.90, 2.05), a una altura de 1.05 metros. Las coordenadas del transmisor son (5.54, 0.60). La colocación del transmisor podría haber sido cualquier otra, pero se eligió esta debido al espacio y accesibilidad que había en ese lugar para colocar los diferentes materiales que se usaron. La potencia de transmisión fue la máxima, 18 dBm.

#### **Caso 1- ARMARIO DE PLÁSTICO:**

-Transmisor situado encima del armario (altura = 0.92 m): Se colocó el transmisor directamente encima del armario de plástico, con visión directa con el receptor. La potencia recibida fue de -37 dBm.

-Transmisor dentro del armario con puertas abiertas (altura = 0.76 m): La potencia recibida fue de -41 dBm.

-Transmisor dentro del armario con puertas cerradas (altura = 0.76 m): La potencia recibida fue de -40 dBm.

Como se observa en las medidas, este tipo de obstáculos no atenúa mucho la señal y no afecta demasiado a la potencia recibida.

#### **Caso 2- CAJÓN DE MADERA:**

Se usó un cajón de madera cuyas paredes laterales eran metálicas, pero el obstáculo en visión directa era la pared de madera.

-Transmisor situado encima del cajón (altura = 0.80 m) = -39 dBm.

-Transmisor situado dentro del cajón cerrado (altura = 0.61 m) = -44 dBm.

La potencia recibida fuera del cajón es prácticamente igual a la recibida en el caso 1 cuando el transmisor también estaba encima del armario de plástico. Pero en este caso, cuando el transmisor está en el interior, la potencia es unos 3 o 4 dBm menor.



### **Caso 3- CAJA DE CARTÓN:**

El transmisor se situó dentro de una caja de cartón convencional, a una altura de 0.92 metros, y la potencia recibida era igual a la potencia recibida sin obstáculo: entre -36 y -37 dBm.

### **Caso 4- CUBIERTA DE PLÁSTICO:**

En este caso el transmisor se cubrió con un recipiente de plástico. La altura también era de 0.92 metros. Los resultados obtenidos en las medidas fueron como los obtenidos con el armario de plástico. Se midieron potencias recibidas entre -41 y -42 dBm.

### **Caso 5- FUNDA DE TELA:**

Para estas medidas se introdujo el transmisor en una funda de tela, tipo porta CDs, y se situó a una altura de 0.92 metros. La potencia recibida fue de -48 dBm, la menor de todos los casos.

En este escenario observamos que las atenuaciones por los materiales descritos anteriormente, aunque no son atenuaciones muy grandes, en muchos casos podrían ser muy importantes a la hora de diseñar redes en escenarios interiores complejos.

Aún así, en este escenario se obtuvo otra información muy útil y más crítica que las atenuaciones introducidas por los materiales: Debido al diagrama de radiación de la antena de nuestros dispositivos XBee-Pro y a la proximidad de algunos de los pequeños cables que componen el circuito de alimentación de los dispositivos, (figuras 6.5 y 6.6) la posición y orientación de los dispositivos es un factor clave a la hora de la transmisión, ya que estando un dispositivo en el mismo lugar, al cambiar la orientación del mismo, la potencia recibida puede llegar a variar hasta 15 dBm.

Los efectos debidos al diagrama de radiación no se pueden cambiar, pero con mantener el dispositivo con la misma orientación para las diferentes medidas, evitamos las diferencias existentes debidas a dicha orientación de la antena.

Las diferencias existentes debidas a las interferencias de los pequeños cables usados en el circuito de alimentación de los dispositivos, se redujeron cambiando el método de alimentación de los dispositivos XBee-Pro. El nuevo método que se usó fue alimentar los dispositivos directamente con un cable USB conectado a un ordenador.



#### 7.2.4.-Escenario 4: Primeras medidas de PER

El escenario escogido para hacer el primer estudio de calidad ha sido el edificio de I+D Jerónimo de Ayaz de la UPNA. Se eligió este edificio debido a que disponíamos de espacios para realizar las medidas y la cantidad de personas que transitan por él es mucho menor que en otros edificios.

Se utilizó el nuevo método con USB para alimentar los dispositivos, por lo que se hizo necesario adquirir un ordenador portátil, ya que para alcanzar los niveles de potencia recibida más bajos, la distancia necesaria entre transmisor y receptor era bastante grande.

Como se muestra en el apartado 8.1.2, se puede elegir el número de paquetes a transmitir para calcular el PER. Para que los valores obtenidos del PER sean significativos, cuanto mayor es el número de paquetes enviados, mejores serán los valores obtenidos. Para este primer estudio, y debido a que para cada medida se necesitaban alrededor de 20 minutos, se decidió transmitir 100.000 paquetes.

La potencia transmitida fue de 18 dBm, y se desactivaron los ACK, por lo que los datos obtenidos son los del peor caso posible.

El transmisor se situó en el laboratorio L5, conectado mediante un cable USB a un ordenador. El receptor se conectó de la misma manera a un ordenador portátil, ya que para muchos de los casos que se necesitaban medir, la distancia era muy grande y se necesitaba un dispositivo que se pudiera mover por todo el edificio.

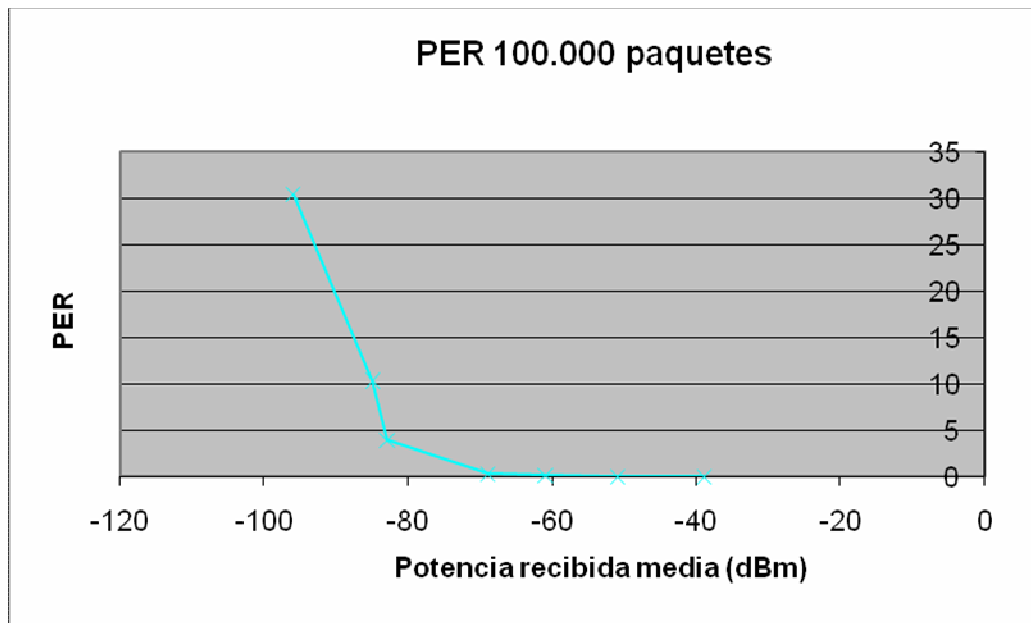
La metodología seguida para realizar las medidas fue la siguiente: Primero se puso el receptor escuchando para ver la potencia que recibía, y a continuación se buscó el lugar donde se daba la potencia que se quería medir. Dejando el receptor en el lugar adecuado, se realizaba la medida del PER. Pasado el tiempo necesario para transmitir los 100.000 paquetes, el programa daba el valor de la potencia media de los paquetes recibidos, el número de paquetes recibidos y el PER. En la tabla 7.11 se muestran los valores obtenidos. La primera columna muestra los valores de potencia recibida que se quería medir, pero debido a que el programa da el valor medio de las potencias recibidas de todos los paquetes, no coincide exactamente.



Potencia (dBm)	Pot. recibida media (dBm)	Paquetes recibidos	PER
-40	-39	99.998	0,002
-50	-51	99.998	0,002
-60	-61	99.875	0,125
-70	-69	99.747	0,253
-80	-83	96.051	3,949
-90	-85	89.671	10,329
-100	-96	69.608	30,392

**Tabla 7.11.** Valores de PER para 100.000 paquetes transmitidos.

En la figura 7.27 se representan los valores obtenidos de PER frente a la potencia media recibida. Se ve claramente que para valores menos de -70 dBm el valor del PER incrementa exponencialmente, y que para valores mayores se pierden muy pocos paquetes.



**Figura 7.27.** Representación de PER para 100.000 paquetes transmitidos.

Cabe destacar que cuanto mayor era la distancia entre transmisor y receptor, más debería afectar el tránsito de personas. Este efecto se estudiará más adelante.

### 7.2.5.-Escenario 5: Mejoras en medidas de PER

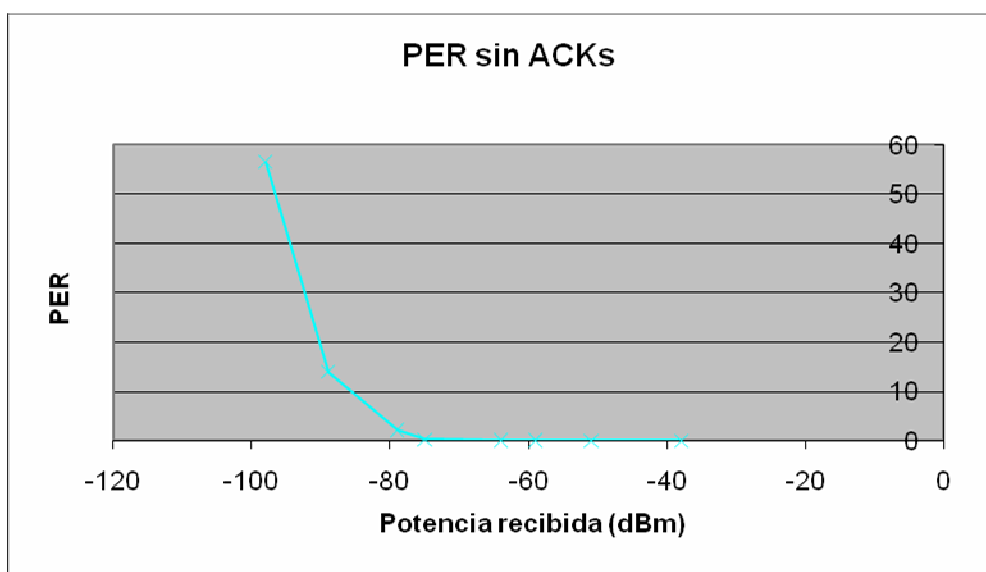
En el escenario anterior se vio que los valores de PER empiezan a crecer mucho a partir de uno -60 dBm. En este nuevo escenario, el lugar para realizar la mayoría de las medidas fue el mismo, el laboratorio L5 del edificio Jerónimo de Ayanz. Aunque para alcanzar niveles de potencia recibida bajos, hubo que situar el receptor fuera del laboratorio.

Para que los datos sean más fiables, en este caso se transmitieron 500.000 paquetes en vez de 100.000, aumentando el tiempo necesario para cada medición hasta 1 hora. El método de medida fue exactamente igual al explicado en el Escenario 4. Pero en este escenario se realizaron medidas sin ACK y otras con ACK, para estudiar las diferencias que deberían existir. En la tabla 7.12 se muestran los datos obtenidos para las medidas sin ACK.

Potencia (dBm)	Pot. recibida media (dBm)	Paquetes recibidos	PER
-40	-38	499.972	0,0056
-50	-51	500.000	0
-60	-59	499.392	0,1216
-65	-64	499.379	0,1242
-75	-75	499.092	0,1816
-80	-79	488.999	2,2002
-90	-89	430.602	13,8796
-100	-98	218.192	56,3616

**Tabla 7.12.** Valores de PER para 500.000 paquetes transmitidos.

En la siguiente gráfica salen representados los datos de la tabla 7.12.



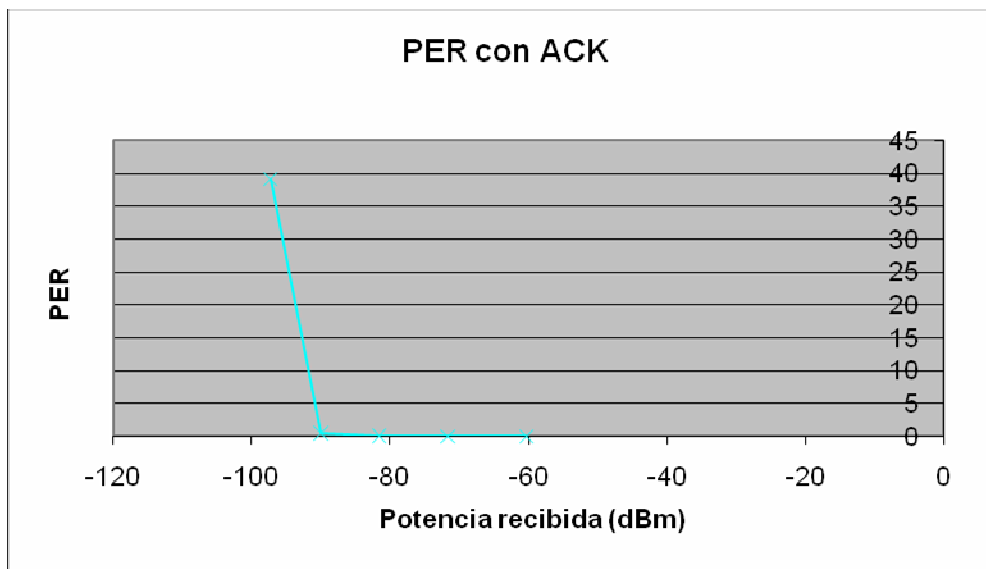
**Figura 7.28.** Representación de PER para 500.000 paquetes transmitidos.



A continuación se programaron los dispositivos para transmitir con 1 ACK y se volvieron a repetir las medidas. Como se aprecia en la tabla 7.13, los resultados mejoran considerablemente. La figura 7.29 muestra estos datos representados en una gráfica.

Potencia (dBm)	Pot. recibida media (dBm)	Paquetes recibidos	PER
-40	-39,4	500.000	0
-50	-51	500.000	0
-60	-60,32	500.000	0
-70	-71,70	499.978	0,0044
-80	-81,6	499.668	0,0664
-90	-90	497.785	0,4430
-100	-97,26	305.216	38,9568

**Tabla 7.13.** Valores de PER para 500.000 paquetes transmitidos con ACK.

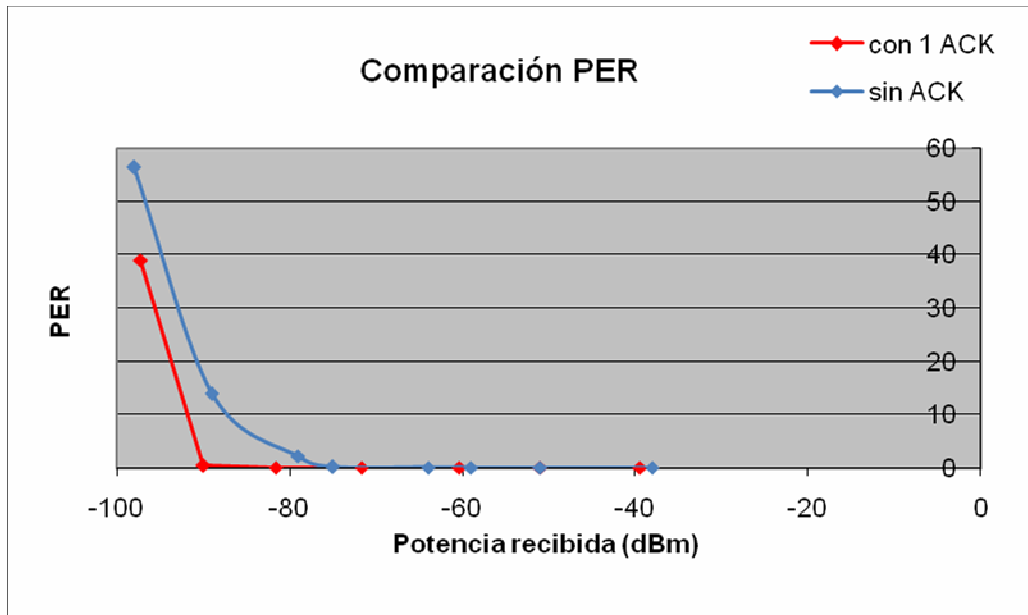


**Figura 7.29.** Representación de PER para 500.000 paquetes transmitidos con ACK.

Al igual que en el escenario anterior, la interacción de las personas altera los resultados, pero se ve claramente la mejoría lograda. El PER empieza a crecer significativamente alrededor de los -90 dBm, y teniendo en cuenta que la sensibilidad es de -100 dBm, podemos afirmar que la transmisión es buena casi en la totalidad de las situaciones. Cabe destacar que esta mejora ha sido sólo con un único ACK, por lo que si se aumenta la cantidad de ACKs, seguiría mejorando la calidad del enlace.



Por último, en la figura 7.30 se representan conjuntamente las dos curvas anteriores para poder compararlas mejor.



**Figura 7.30.** Comparación de PER para 500.000 paquetes transmitidos sin ACK y con un ACK.

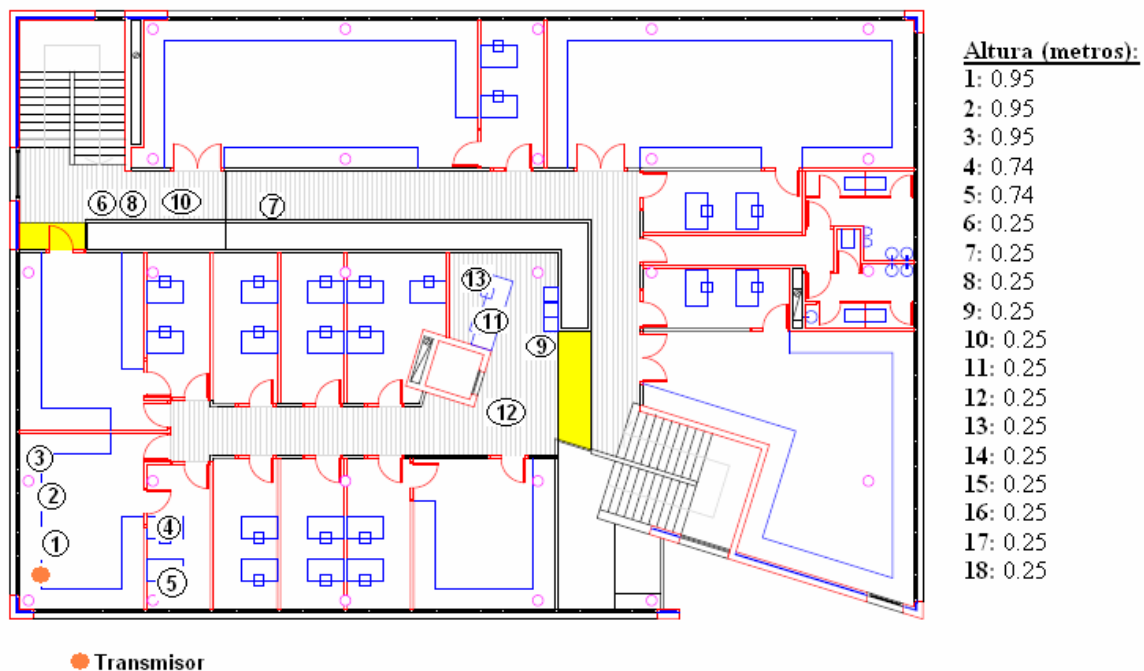


### 7.2.6.-Escenario 6: PER con 1.000.000 paquetes transmitidos

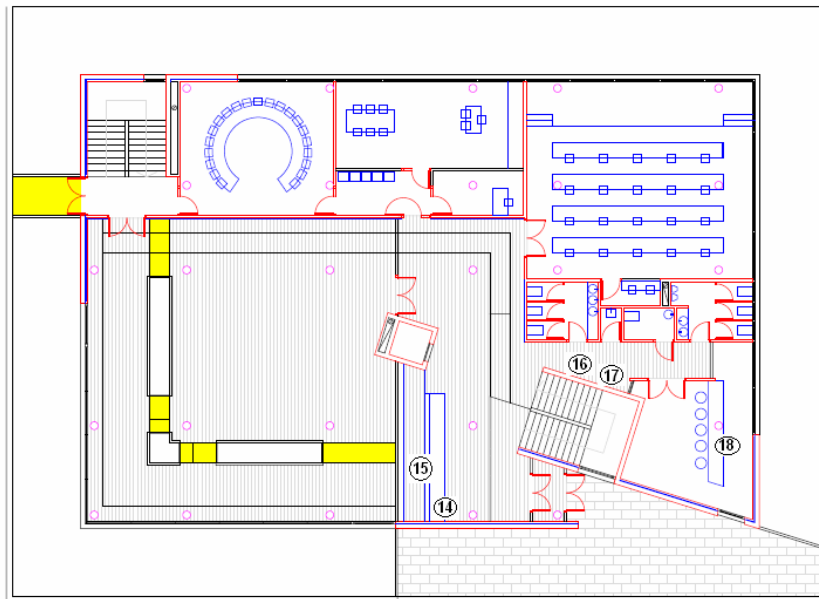
Ya hemos visto en los dos escenarios anteriores cómo empeora el PER al disminuir la potencia recibida. Pero para obtener datos para enviar a congresos, los datos obtenidos deben ser lo mejor posibles. Por ello, en este escenario se volvieron a realizar medidas de PER, pero con 1.000.000 de paquetes transmitidos.

El escenario interior para realizar las medidas fue el mismo que en los casos anteriores, el edificio de la UPNA Jerónimo de Ayanz. El método de medida fue el mismo que en el escenario anterior. La única diferencia fue que el tiempo requerido para cada obtención de datos aumentó a 2 horas debido al incremento de los paquetes a enviar.

Todas las medidas realizadas en este escenario fueron sin ACK, y por uniformidad, con potencia transmitida de 18 dBm para todos los casos. Debido a que era necesario alcanzar niveles de potencia recibida muy bajos, hasta llegar a la sensibilidad de los dispositivos, muchas de las medidas fueron realizadas fuera del laboratorio L5, lugar donde se habían realizado la mayor parte de las medidas en los escenarios anteriores. Por tanto, muchas medidas se realizaron por todo el edificio Jerónimo de Ayanz, en algunos casos teniendo que cambiar de planta. Las posiciones exactas del receptor en cada caso se muestran en las figuras 7.31 y 7.32. El transmisor se situó en el laboratorio L5, en las coordenadas (1.40, 0.80), a una altura de 0.88 metros.



**Figura 7.31.** Plano de la primera planta con posiciones para PER de 1.000.000 paquetes transmitidos.



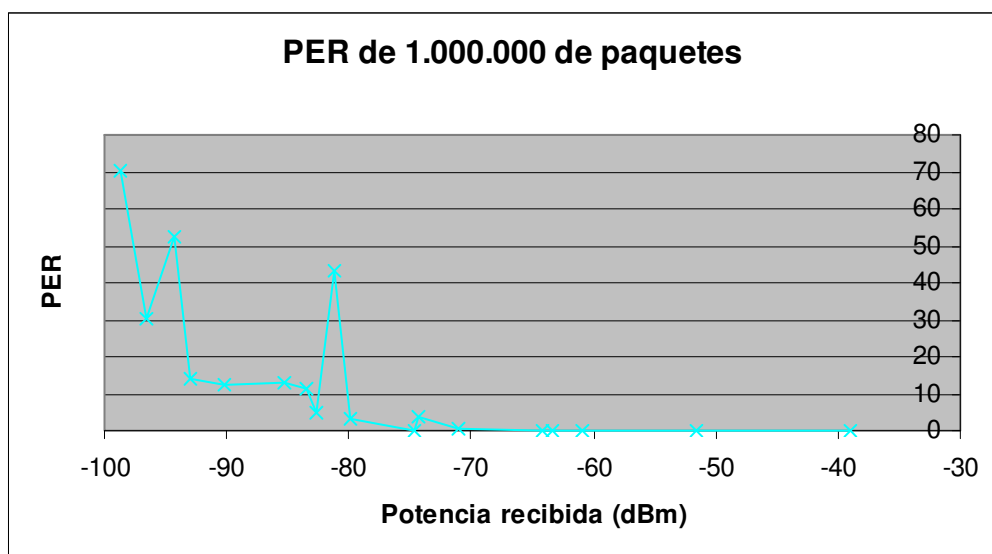
**Figura 7.32.** Plano de la planta baja con posiciones para PER de 1.000.000 paquetes transmitidos.

Como se ha visto en los casos anteriores, los valores donde el PER empieza a empeorar exponencialmente son los datos más significativos para el estudio, por lo que en este escenario se realizaron muchas más medidas que en los anteriores, sobre todo para potencias recibidas menores de -80 dBm, para tener mayor cantidad de datos en la zona de interés. En la tabla siguiente se muestran todos los datos obtenidos en estas medidas:

Caso	Pot. recibida media (dBm)	Paquetes recibidos	PER
1	-39	999.942	0,0058
2	-51,65	999.903	0,0097
3	-60,92	997.452	0,2548
4	-63,34	997.662	0,2338
5	-64,16	998.562	0,1438
6	-71,07	993.725	0,6275
7	-74,39	960.837	3,9163
8	-74,65	998.578	0,1422
9	-79,95	968.453	3,1547
10	-81,18	568.696	43,13
11	-82,64	953.520	4,648
12	-83,41	884.840	11,516
13	-85,31	872.529	12,7471
14	-90,11	876.477	12,35
15	-92,97	858.610	14,139
16	-94,26	475.344	52,46
17	-96,54	698.264	30,1736
18	-98,75	294.845	70,52

**Tabla 7.14.** Valores de PER para 1.000.000 paquetes transmitidos.

En la figura 7.33, que se muestra a continuación, se observa una gráfica con los datos de la tabla 7.14.



**Figura 7.33.** Representación de PER para 1.000.000 paquetes transmitidos.

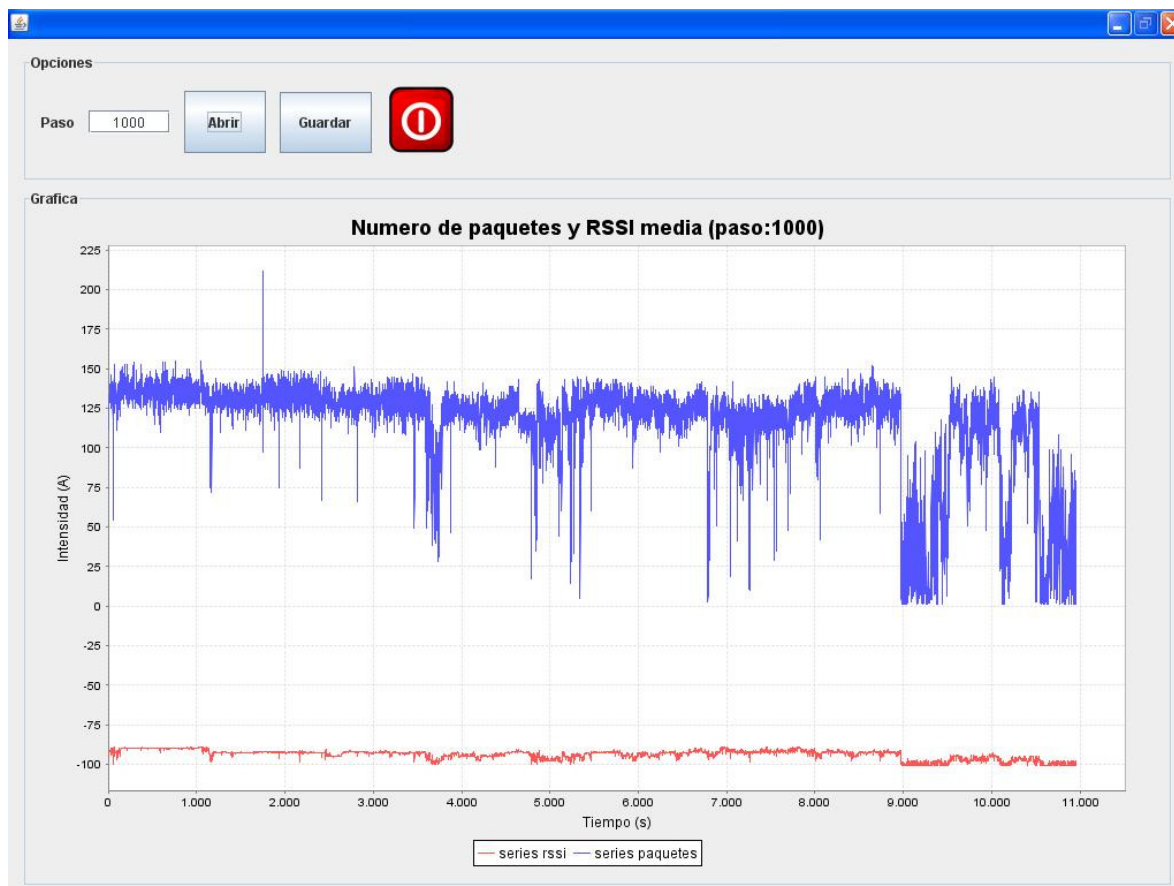
En la gráfica se aprecia que los datos obtenidos siguen la misma tendencia exponencial que en los escenarios con 100.000 y 500.000 paquetes transmitidos. Pero en este caso, que se han tomado muchos más datos, se pueden distinguir claramente que algunos de ellos se salen de la línea exponencial. En estos casos, el PER medido es mucho mayor de lo esperado.

Debido a que estas medidas tardaban 2 horas en realizarse, se pensó que la diferencia de horarios a la hora de realizar las mediciones podría ser un factor clave, ya que las condiciones de las medidas realizadas a primera hora de la mañana no eran las mismas que las que se hacían a continuación. A primera hora de la mañana, sobre las 7.00, el edificio se encontraba totalmente vacío, y al ir transcurriendo la mañana, se iba llenando de gente, con el consiguiente efecto que esto puede tener en la señal. Además, para las medidas con potencia recibida baja, el receptor se situaba en la planta baja, donde el tránsito de personas era mucho mayor.

Por tanto, se creó un programa parecido a los usados para obtener los datos de PER y RSSI, pero con el que se obtenía una evolución temporal de la señal recibida. De esta manera, se podría hacer una comparativa de cómo variaba la señal durante varias horas. El código del programa se puede consultar en el anexo B.

Para ello, se situó un transmisor en el laboratorio L5, y un receptor en la planta baja, concretamente en la posición 15 marcada en el plano de la figura 7.32. Primero se realizó una medida de 1.000.000 de paquetes transmitidos, la cual empezó a las 7.00 y terminó a las 9.00. Nada más terminar esta medida, se

guardaron los datos obtenidos y con el transmisor y receptor en la misma posición, se volvió a hacer otra medida de 1.000.000 de paquetes, desde las 9.00 hasta las 11.00 aproximadamente. En la figura 7.34 se muestra una captura de pantalla del programa con los datos obtenidos.



**Figura 7.34.** Evolución temporal de RSSI y paquetes recibidos.

Debido a la cantidad de datos, y para una mejor comprensión, el programa tiene una opción llamada 'Paso' donde se puede seleccionar el tiempo en milisegundos que se analizan para obtener un valor representable. Por ejemplo, en la figura anterior el paso es de 1000, por lo que los datos que se representan son de los paquetes recibidos cada 1 segundo. De color azul se representa el número de paquetes que se han recibido en ese tiempo, y de color rojo se representa la media de la potencia de esos paquetes recibidos. Se ve claramente cómo al avanzar la mañana, tanto el nivel de la potencia recibida y la cantidad de paquetes recibidos va disminuyendo. Se intentó colocar el receptor funcionando lo más cerca posible del nivel de sensibilidad, para que los posibles cambios fueran más evidentes.

Con estos resultados podemos confirmar las sospechas, y cuanto mayor es la cantidad de movimiento de personas que había dentro del edificio, mucho menor era el número de paquetes que se recibían, acentuándose el problema para niveles bajos de potencia recibida.

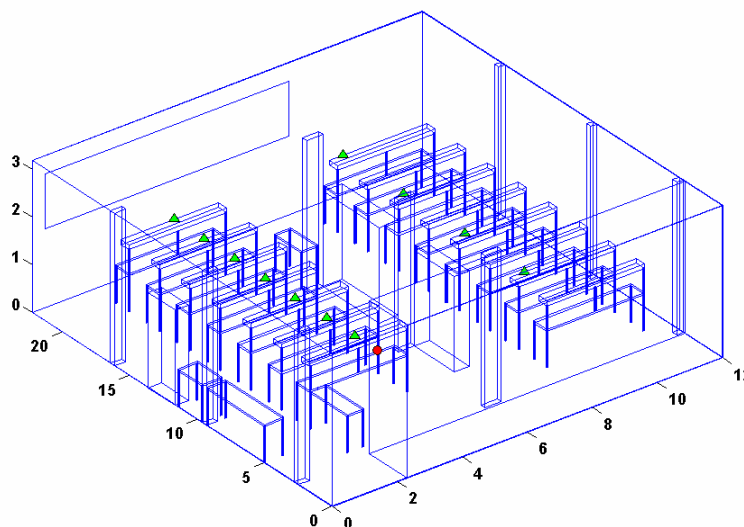
### 7.2.7.-Escenario 7: RSSI, PER y trazado de rayos

Una vez analizados y desarrollados los métodos para medir el PER y RSSI, en este nuevo caso se realizaron las mismas medidas, pero cambiando de escenario. Este nuevo escenario fue el laboratorio de Radiocomunicación de la Universidad Pública de Navarra. En él se incluyen multitud de elementos como mesas, sillas, armarios y una gran cantidad de instrumentos. La figura 7.35 muestra una imagen del escenario.



**Figura 7.35.** Laboratorio de Radiocomunicación, UPNA.

La razón de esta elección fue que compañeros de la UPNA habían realizado simulaciones de trazado de rayos para este laboratorio anteriormente. Por tanto, recopilar más información mediante medidas reales era una tarea interesante para poder comparar resultados y obtener un análisis más completo. En la figura 7.36 se muestra un esquemático del trazado de rayos.



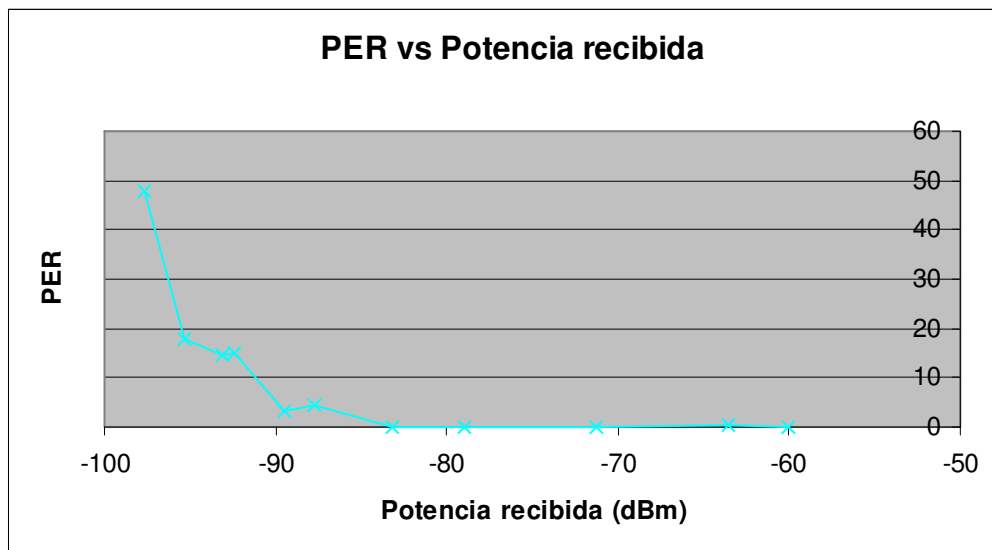
**Figura 7.36.** Esquemático del trazado de rayos 3D del laboratorio de Radiocomunicación de la UPNA.



Para la realización de las medidas, el método seguido fue el mismo. Se colocó un único transmisor y un solo receptor. El transmisor no se movió de su posición en las distintas medidas, mientras que el receptor se colocaba en distintas posiciones dentro del laboratorio dependiendo de la potencia recibida necesaria para cada caso. Se transmitieron para cada medida 1.000.000 de paquetes, sin ACK. Cabe destacar que las medidas se realizaron sin personas dentro del laboratorio, por lo que se minimizó la influencia que este factor tiene en las medidas. La siguiente tabla y la figura 7.37 muestran los resultados de PER frente a potencia recibida obtenidos:

Pot. recibida media (dBm)	Paquetes recibidos	PER
-59,99	998.086	0,1914
-63,54	997.662	0,2338
-71,32	999.677	0,0323
-79	999.472	0,0528
-83,16	999.730	0,027
-87,71	955.983	4,4017
-89,51	968.340	3,166
-92,41	849.528	15,0472
-93,14	855.524	14,4476
-95,32	821.127	17,8873
-97,61	519.802	48,0198

**Tabla 7.15.** Valores de PER para 500.000 paquetes transmitidos con ACK.



**Figura 7.37.** Representación de los datos de la tabla 7.15.

Los datos obtenidos son los esperados. La curva sigue la tendencia exponencial típica vista hasta ahora. El PER empieza a aumentar considerablemente con valores de potencia muy baja. También se observa que debido a la poca interferencia humana dentro del laboratorio, no aparecen en la gráfica los picos que existían en el escenario 6.

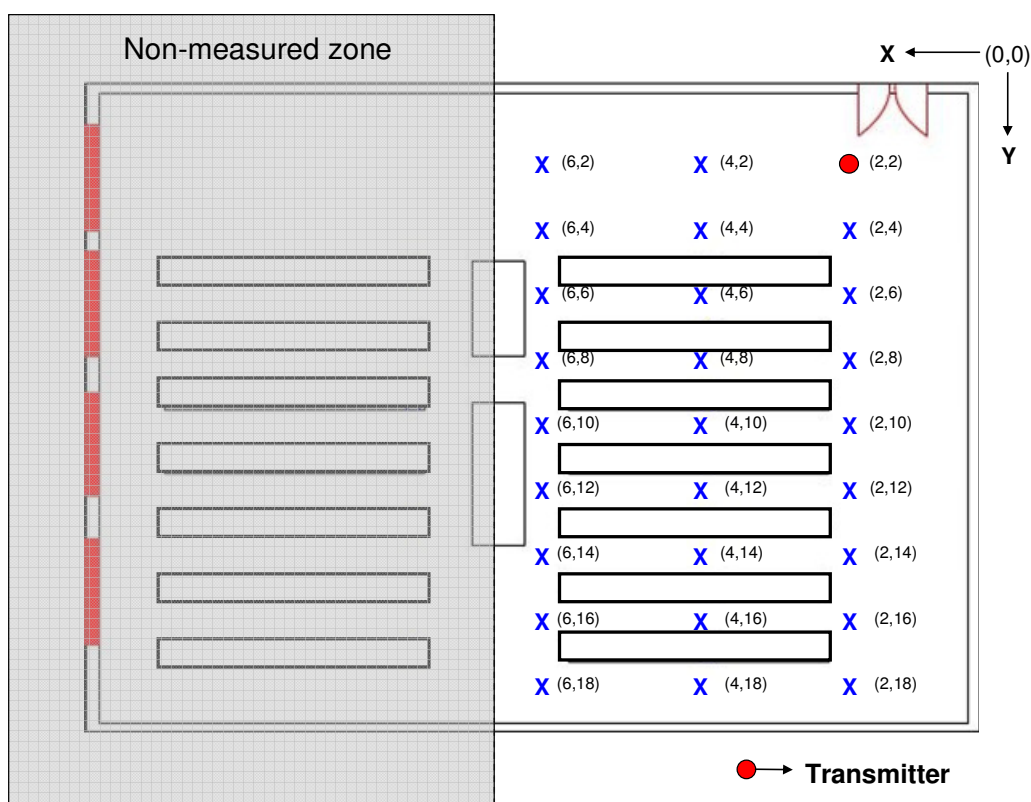
## 7.2.8.-Escenario 8: Dispositivos CyFi. Universidad de Valencia

Durante este trabajo surgió la posibilidad de trabajar conjuntamente con un equipo de investigación de la Universidad de Valencia. Este grupo había desarrollado un grupo de motas CyFi con el fin de ser empleadas en aplicaciones de monitorización y cuya configuración se describió en el apartado 6.2 de este trabajo. El trabajo que se realizó con ellas en la UPNA fue el estudio de la parte radio de dichas motas.

Al igual que con los dispositivos XBee-Pro, se realizaron medidas de RSSI, consumo de corriente y PER.

### 1- Medidas de RSSI:

El escenario elegido para realizar las medidas fue el laboratorio de Radiocomunicación de la UPNA (figura 7.35). En él se hizo un barrido de potencia recibida, situando un único transmisor alimentado por pilas en las coordenadas (2,2), como se muestra en la figura 7.38. El origen de coordenadas es la esquina superior derecha del plano, y las unidades son metros.



**Figura 7.38.** Plano del laboratorio de Radiocomunicación de la UPNA con las posiciones de las medidas.



El receptor, que se iba a ir colocando en las posiciones marcadas en el plano con una 'X' de color azul (figura 7.38), permanecía conectado a un ordenador portátil mediante un cable USB.

Tanto el transmisor como el receptor se colocaron siempre a una misma altura de 0.81 metros. Y para evitar posibles diferencias debidas al diagrama de radiación de la antena, la orientación de los dispositivos en todas las posiciones fue la misma.

Conjuntamente con las motas, se adjuntaba un programa con el que se podían leer los datos de la potencia recibida y el consumo del dispositivo. No se adjuntaba programa para configurar los parámetros de la transmisión.

Los resultados de las primeras medidas fueron los mostrados en la tabla 7.16. Las unidades del valor de las celdas son dBm.

Coordenada Y	Coordenada X		
	6	4	2
2	-19	-19	<b>Tx</b>
4	-15	-18	-18
6	-20	-19	-21
8	-31	-31	-31
10	-31	-31	-31
12	-23	-31	-31
14	-12	-23	-21
16	-24	-21	-17
18	-24	-17	-25

**Tabla 7.16.** Valores de RSSI con motas CyFi.

Hay que añadir que estos datos, proporcionados por el software adjunto, se midieron con una tasa de transmisión de 1 paquete cada 20 segundos. Debido a la imposibilidad de poder cambiar los parámetros de transmisión, se enviaron de vuelta todas las motas para que en Valencia cambiaran la velocidad de transmisión a su máximo valor (supuestamente 250 Kbps).

Cabe destacar que a diferencia de los XBee-Pro, las motas CyFi tienen la capacidad de regular su potencia de transmisión para mantener constante la potencia recibida por los receptores, independientemente de su posición. En la tabla 7.16 se ve claramente cómo al alejarse el receptor del transmisor, la potencia recibida no tiende a caer.

Cuando se recibieron de nuevo las motas, configuradas para transmitir a su máxima tasa, se volvió a realizar el mismo estudio. Como se verá a continuación, los resultados varían mucho con respecto al caso de transmitir un paquete cada 20 segundos. En la tabla 7.17 se pueden ver estos datos.



Coordenada Y	Coordenada X		
	6	4	2
2	-15.5	-14.5	<b>Tx</b>
4	-13	-12.5	-14.71
6	-12*	-13	-19.82
8	-14.36	-12*	-14.4
10	0	0	-11*
12	0	0	0
14	0	0	0
16	0	0	0
18	0	0	0

**Tabla 7.17.** Valores de RSSI con motas CyFi a máxima tasa de transmisión.

A diferencia del caso anterior, a partir de cierta distancia ya no se reciben paquetes. Además, el valor proporcionado por el software variaba mucho, por lo que el valor de la tabla es una media con los paquetes llegados en un tiempo de 2 segundos (alrededor de 100 paquetes). El asterisco indica que esos valores medios se calcularon con menos de 10 paquetes. Se ve claramente que al aumentar la velocidad de la mota diseñada a su valor máximo, las prestaciones bajan, ya que el alcance es muy limitado.

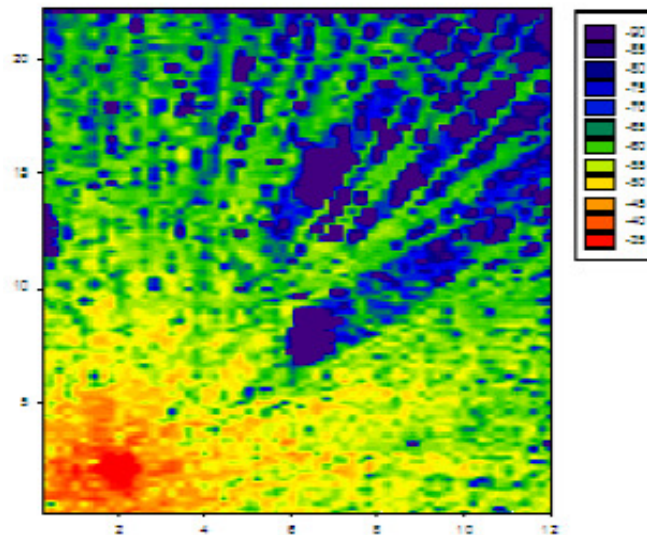
Teniendo en cuenta los problemas para poder hacer una correcta medida de la RSSI en estas condiciones, y añadiendo que los datos de consumo que indicaba el mismo software no variaba de 85 (tampoco indicaba unidades), se decidió crear un programa para leer los datos directamente desde la mota. A este programa se le llamó 'ZigBeePERReceiverV', y se puede consultar su código en el anexo B.3. En la figura siguiente se puede ver una captura de pantalla de dicho programa:



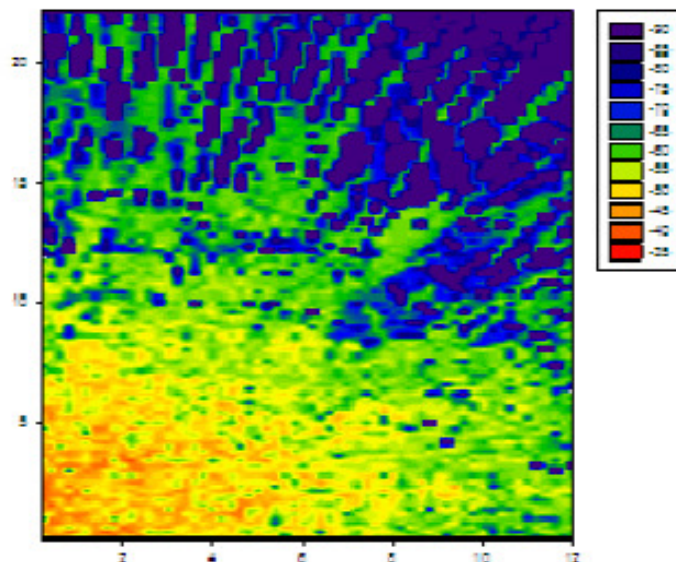
**Figura 7.39.** Software 'ZigBeePERReceiverV'.

Todos los datos obtenidos de aquí en adelante fueron con este nuevo software.

También se obtuvieron resultados de simulación de trazado de rayos 3D para todo el volumen del laboratorio. La antena transmisora se configuró con potencia y parámetros de transmisión equivalentes a un sistema ZigBee convencional. Las figuras 7.40 y 7.41 representan el nivel de potencia recibida en los planos bidimensionales a 0.81m de altura (la altura en la que se tomaron las medidas anteriormente) y a 2.3m respectivamente.



**Figura 7.40.** Resultados de simulación (RSSI) del trazado de rayos 3D para el plano bidimensional a 0.81m de altura.

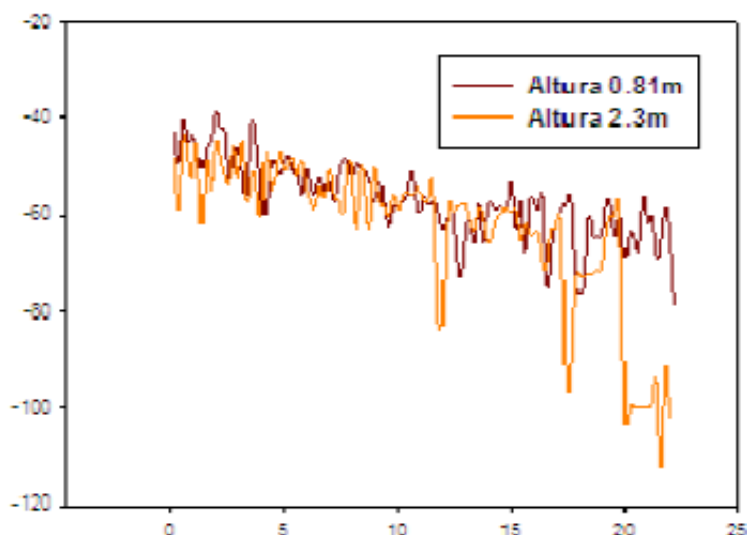


**Figura 7.41.** Resultados de simulación (RSSI) del trazado de rayos 3D para el plano bidimensional a 2.3m de altura.

Como se puede observar en las dos figuras anteriores, el nivel de potencia recibida depende fuertemente de la posición de los elementos receptores. Las variaciones pueden ser del orden de 10 dB en 1 metro, lo que tiene un gran impacto en el funcionamiento de los sensores, no sólo en los límites de

sensibilidad de los receptores, también en la capacidad global del sistema, que depende del nivel de señal y de la relación señal-ruido (SNR).

La figura 7.42 muestra los resultados de simulación, para las dos alturas previamente mostradas, a lo largo de una radial (eje x) en el escenario. La variación de la señal es debida al efecto de las componentes multirayecto típicas en un entorno interior complejo.



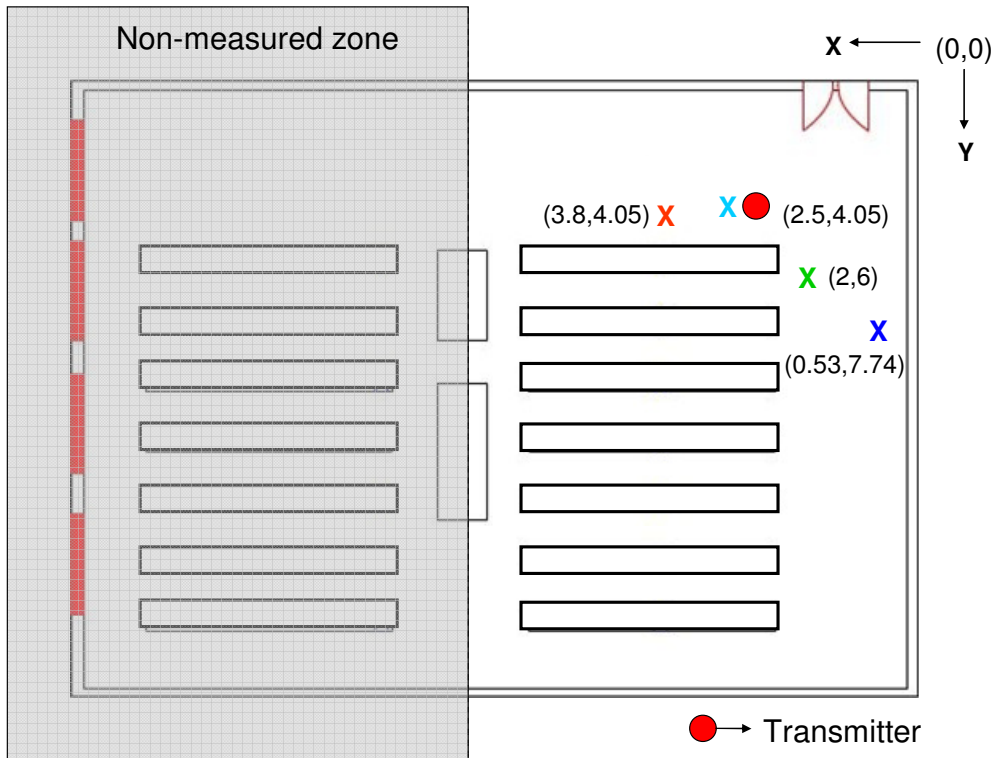
**Figura 7.42.** Resultados de simulación para las alturas 0.81m y 2.3m, a lo largo del eje x del escenario interior analizado.

## **2- Medidas de consumo:**

Para las medidas de consumo, se ha utilizado el mismo método que en escenarios anteriores. Es decir, introducir la resistencia de  $1 \Omega$  en paralelo y medir con el osciloscopio.

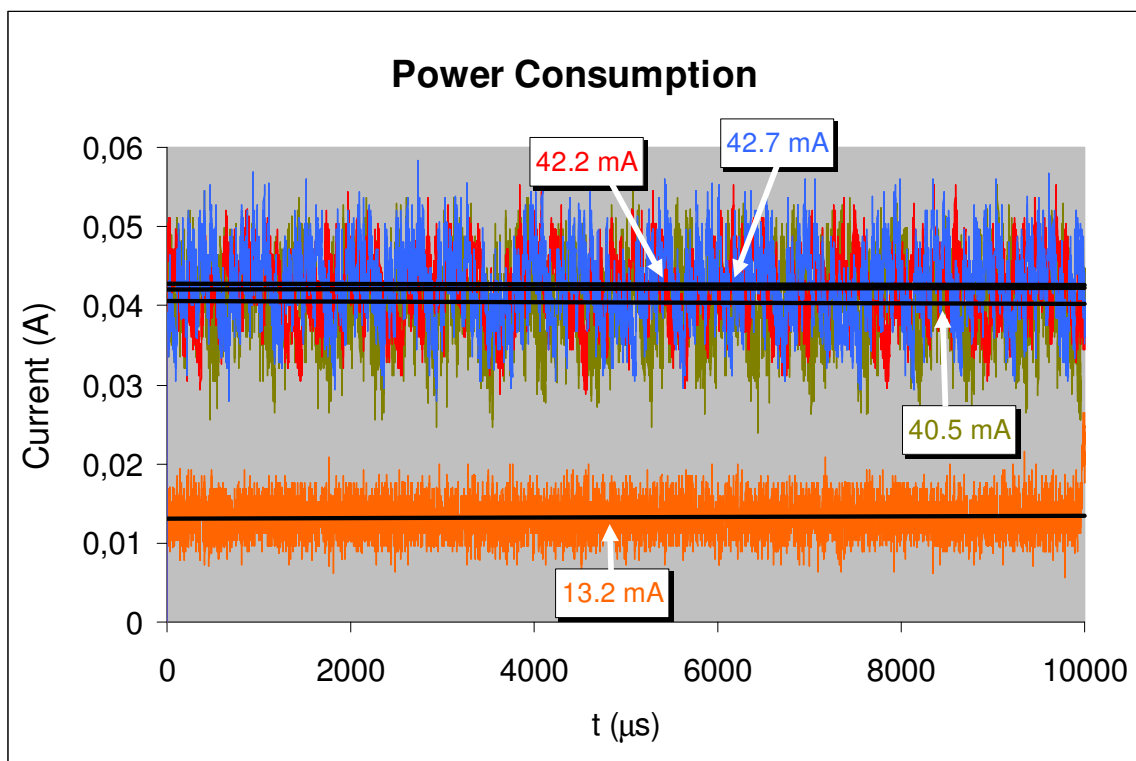
El escenario de medida sigue siendo el laboratorio de Radiocomunicación de la UPNA. El transmisor es alimentado por pilas, y los receptores mediante conexión USB a un ordenador. La altura para todos los dispositivos es de 0.81 metros.

Como ya se ha comentado con anterioridad, a diferencia de los dispositivos XBee-Pro, los CyFi autorregulan la potencia de transmisión dependiendo de la distancia a que se encuentre el receptor. Por tanto, se decidió realizar las medidas del consumo de corriente del transmisor en cuatro situaciones distintas, tanto en standby como transmitiendo, para comprobar que el consumo cambiaba. En la figura 7.43 se muestra la localización del transmisor y los receptores para cada una de las cuatro situaciones.

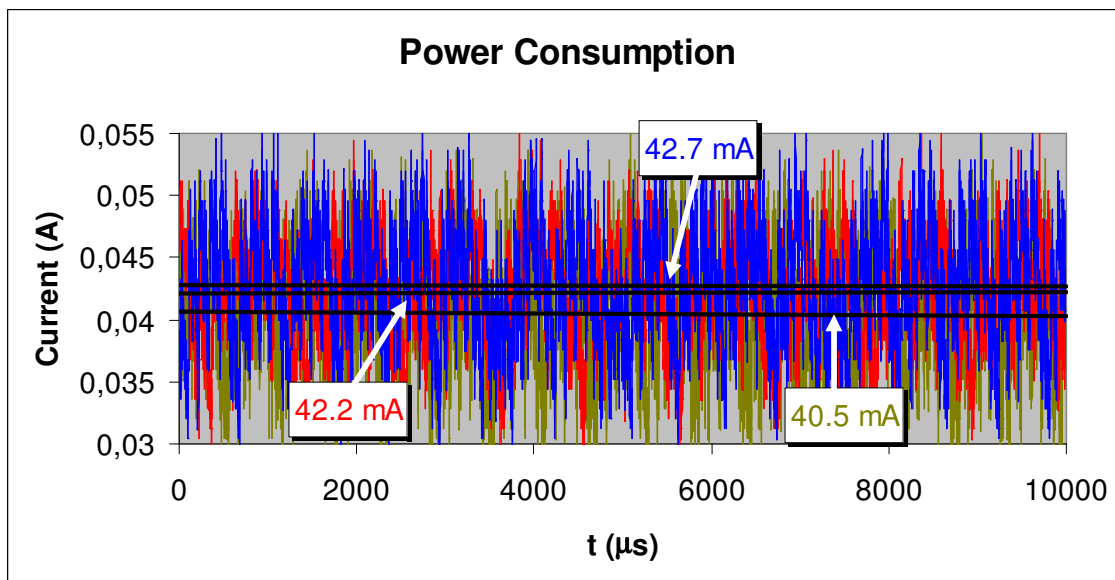


**Figura 7.43.** Plano del escenario con las cuatro posiciones para la medida del consumo de las motas CyFi.

Las medidas han sido realizadas estableciendo un elemento coordinador y un sensor inalámbrico. Las motas han sido programadas para proporcionar alta tasa de transmisión de paquetes, por lo que el alcance es muy limitado, y la 'X' de color azul oscuro de la figura 7.43, es el punto donde se haya el límite de la sensibilidad. Los resultados se pueden observar en la figura 7.44 y la figura 45.



**Figura 7.44.** Variación en el consumo de potencia de las motas en función de la posición. La curva inferior (13.2 mA) es para el modo en standby. El resto de las curvas van desde la posición más cercana a la más lejana.



**Figura 7.45.** Detalle del consumo de potencia de la mota en el escenario interior. Los valores medios de las corrientes son proporcionales a la separación entre motas.

Como puede apreciarse en la figura 7.44, existe una clara distinción entre el modo standby y el modo en transmisión, la cuál era de esperar debido al



normal procedimiento de operación en un transceptor inalámbrico, cuando la placa de RF está en modo activo o en modo idle.

En la figura 7.45, se presentan diferentes valores medios de corrientes, obtenidos para diferentes posiciones del sensor dentro del escenario. Al aumentar la distancia, el nivel de consumo de potencia también crece, como era de esperar debido al control automático de ganancia. Por tanto, al aumentar la distancia y (en principio) disminuir el nivel de potencia recibido, mayor es la demanda de potencia del receptor, incrementando el nivel de la señal radio transmitida, y al mismo tiempo, el consumo de corriente.

### **3- Medidas del PER:**

Se realizó una medida para calcular el PER, con el transmisor en las coordenadas (2,6) y un receptor en las coordenadas (2,2) (ver figura 7.43).

Se transmitieron 100.000 paquetes, sin ACK, de los que solamente llegaron 1.363. Por tanto, el PER obtenido fue de 98,637.

Teniendo en cuenta que no es el punto del límite de sensibilidad, es un valor muy malo, inviable para muchas aplicaciones.

Por todos los resultados obtenidos con los dispositivos CyFi, se deduce que para la tasa de transmisión máxima, las prestaciones son muy deficientes, aunque hay que tener en cuenta que muchas aplicaciones posiblemente requieran velocidades de transmisión más bajas, puesto que transmitirán muy de vez en cuando. Lo que se traducirá en una mejoría considerable en las prestaciones de las motas.

### 7.2.9.-Escenario 9: Interferencia WiFi

Tanto este escenario como los dos siguientes, son diferentes líneas de trabajo que se comenzaron a realizar, pero que no ha habido tiempo para desarrollar. En todo caso, se presentarán a continuación, siendo un punto de partida útil para futuros estudios.

En este escenario se comenzó a estudiar el posible efecto que pueden provocar las redes WiFi en una red ZigBee, ya que ambas tecnologías trabajan principalmente en la misma banda de frecuencias. Para comenzar, se utilizaron los analizadores de espectro existentes en el laboratorio de Radiocomunicación de la UPNA para visualizar los espectros de la red WiFi y de los dispositivos CyFi. Estos analizadores son el modelo R3131 de ADVANTEST [10]. Los resultados obtenidos se muestran en las figuras 7.46 y en la figura 7.47. En la primera se puede ver el espectro de los dispositivos CyFi, configurados como se recibieron de Valencia. En la segunda figura se muestran simultáneamente los espectros de las motas CyFi y de la red WiFi de la UPNA.

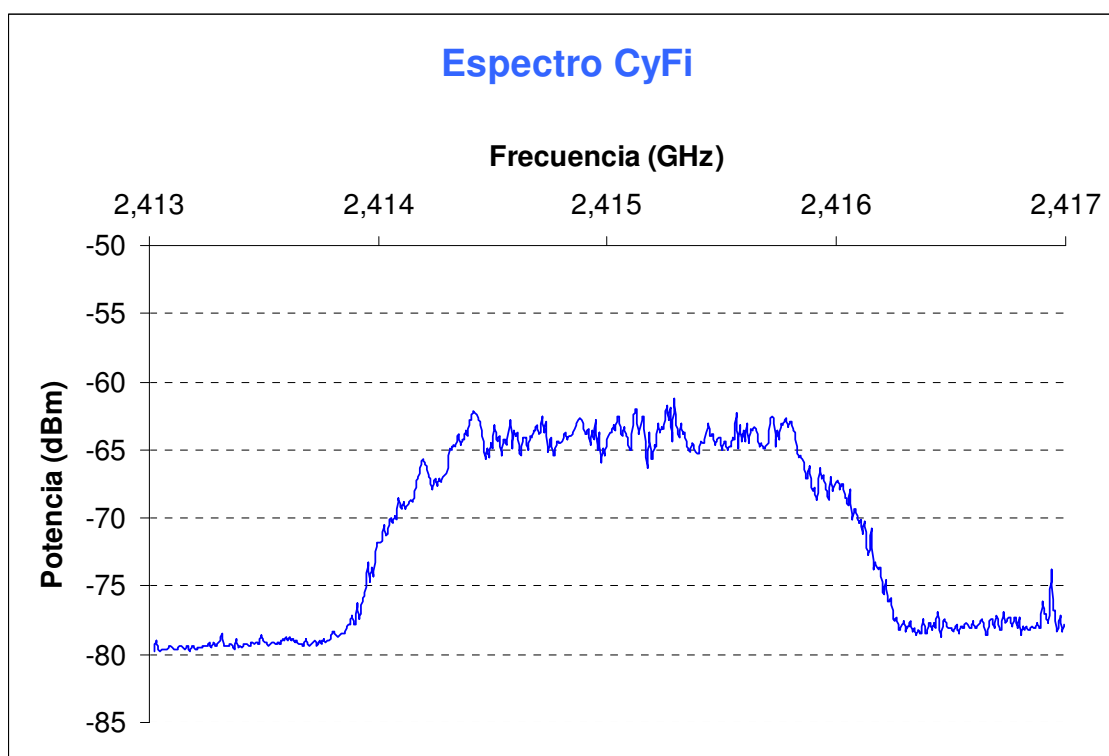
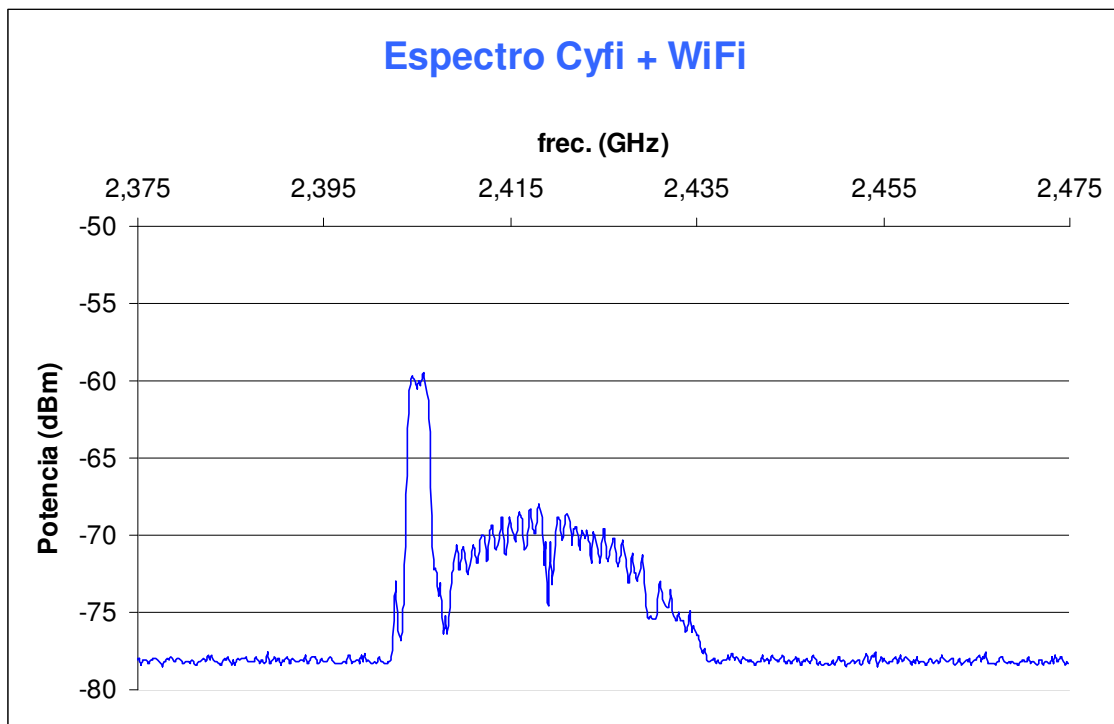


Figura 7.46. Espectro de las motas CyFi.



**Figura 7.47.** Espectro de las motas CyFi, a la izquierda. Espectro de la red WiFi de la UPNA, a la derecha.

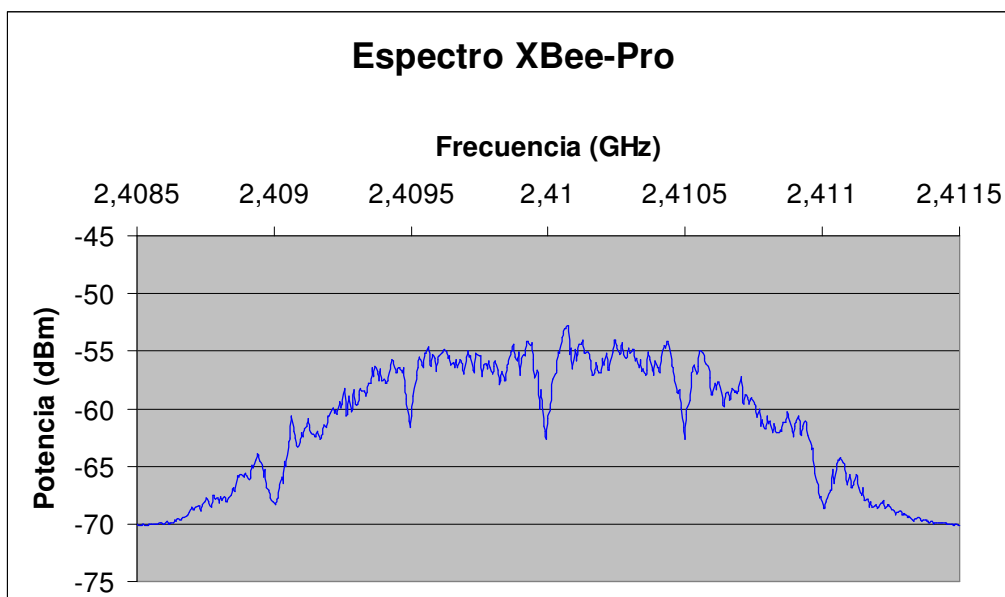
Se ve claramente cómo el nivel del espectro de las motas es mucho mayor que el de la red WiFi. Además, la parte del espectro de WiFi que se solapa con el de las motas es de un nivel mucho menor. Por tanto, si la transmisión CyFi se da en este canal, se presupone que no afectará a la calidad de la red, aunque si el canal CyFi fuera otro que cayera en medio del espectro WiFi, podría haber problemas.

Como ya se ha dicho, el estudio más profundo estas interferencias queda para el futuro.

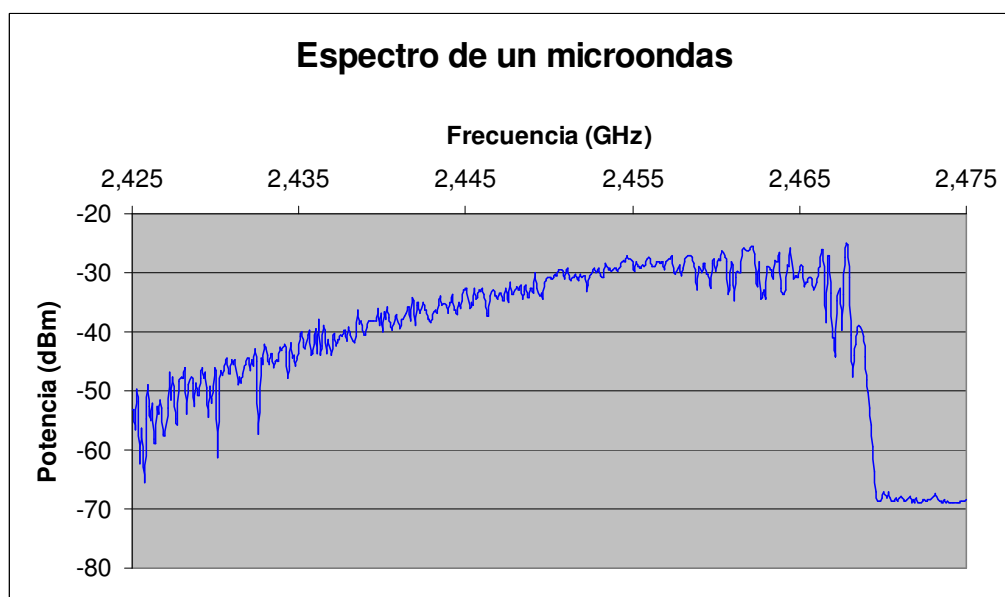


### 7.2.10.-Escenario 10: Interferencia de un microondas

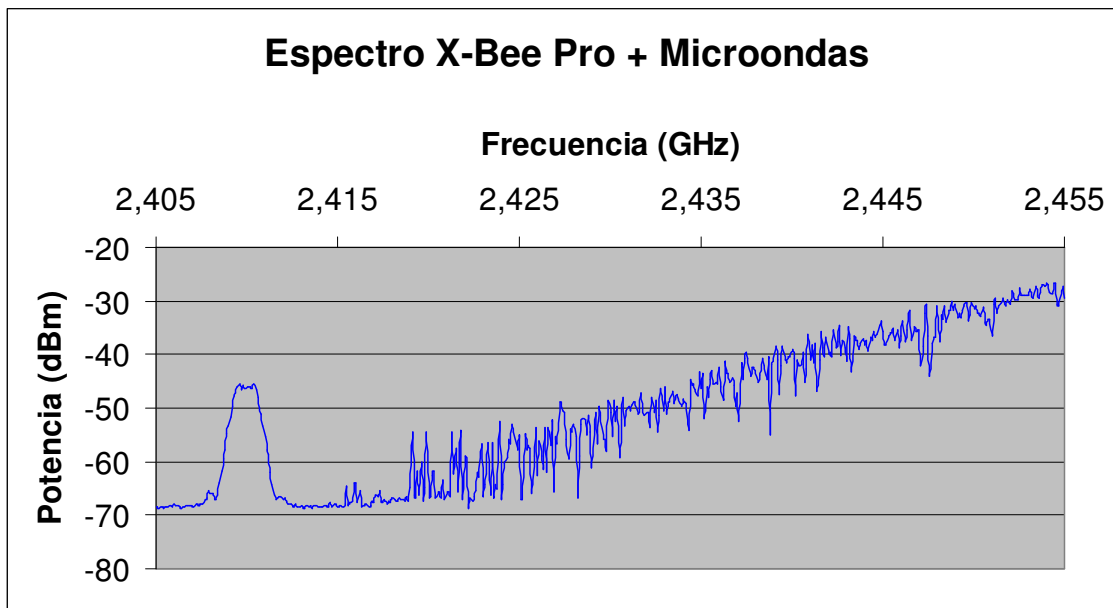
Al igual que en el escenario anterior, aquí también se quiere estudiar el impacto que puede tener la cercanía de un electrodoméstico como el microondas en un dispositivo ZigBee. En este escenario se utilizaron los dispositivos XBee-Pro, transmitiendo a su nivel máximo de potencia, 18 dBm. En la figura 7.48 se puede ver el espectro del XBee-Pro, situado a 10 cm de la antena del analizador de espectros. En la figura 7.49 se muestra el espectro de un microondas común, también situado a 10 cm de la antena del analizador. Por último, en la figura 7.50 se observan simultáneamente ambos espectros.



**Figura 7.48.** Espectro de los dispositivos XBee-Pro transmitiendo a máxima potencia.



**Figura 7.49.** Espectro de un microondas común.



**Figura 7.50.** Espectro del XBee-Pro, en el canal 12 del estándar ZigBee (pico de la izquierda). Espectro de un microondas común, a la derecha.

En el canal número 12, que es en el que se está transmitiendo, se ve que la cercanía de un microondas no afectaría. Pero los demás canales de la banda de 2.4 GHz del estándar ZigBee, exceptuando el número 11, están situados a frecuencias contiguas y mayores que el canal 12 (hasta el 26, ver figura 5.9). Por lo que se ve claramente que la cercanía de un microondas, en la mayoría de los canales en que transmite un dispositivo ZigBee, podría encubrir totalmente la señal ZigBee, imposibilitando la comunicación.

Al igual que el escenario anterior, un estudio más profundo se deja abierto para el futuro.

### 7.2.11.-Escenario 11: Movilidad

Hasta aquí, todos los estudios se han realizado en escenarios estáticos. En este último escenario se comenzó a estudiar cómo afecta a los dispositivos ZigBee un escenario dinámico. Por tanto, entrarán en escena nuevos fenómenos que antes no se tenían en cuenta, como el efecto Doppler. No se entrará en detalles, pero en la práctica, el efecto Doppler dará lugar a una degradación y ensanchado espectral de la señal original.

El estudio de la movilidad se realizó entre un dispositivo transmisor colocado en un coche y un receptor estático situado a 5 metros de la carretera, a una altura de 0,25 metros. La instalación del dispositivo ZigBee transmisor se realizó en el retrovisor derecho del vehículo (ver figura 7.51), quedando al lado de donde se colocó el receptor, evitando que el vehículo fuera un obstáculo directo.



**Figura 7.51.** Detalle del dispositivo transmisor.

El escenario donde se desarrolló el trabajo fue la carretera cercana al edificio de I+D de la Universidad Pública de Navarra (ver figura 7.52). De color azul claro está representada la trayectoria seguida por el vehículo, de arriba a abajo. De color rojo está marcada la posición del receptor. En las figuras 7.53 y 7.54 se pueden ver fotos del escenario a pie de calle.

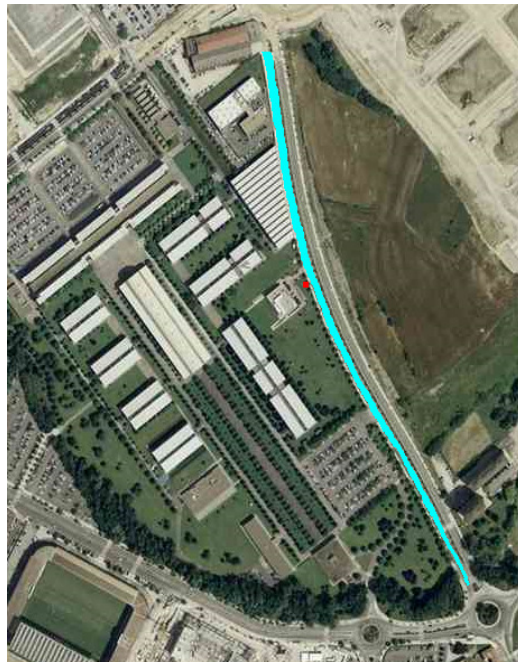


Figura 7.52. Imagen de satélite del escenario.

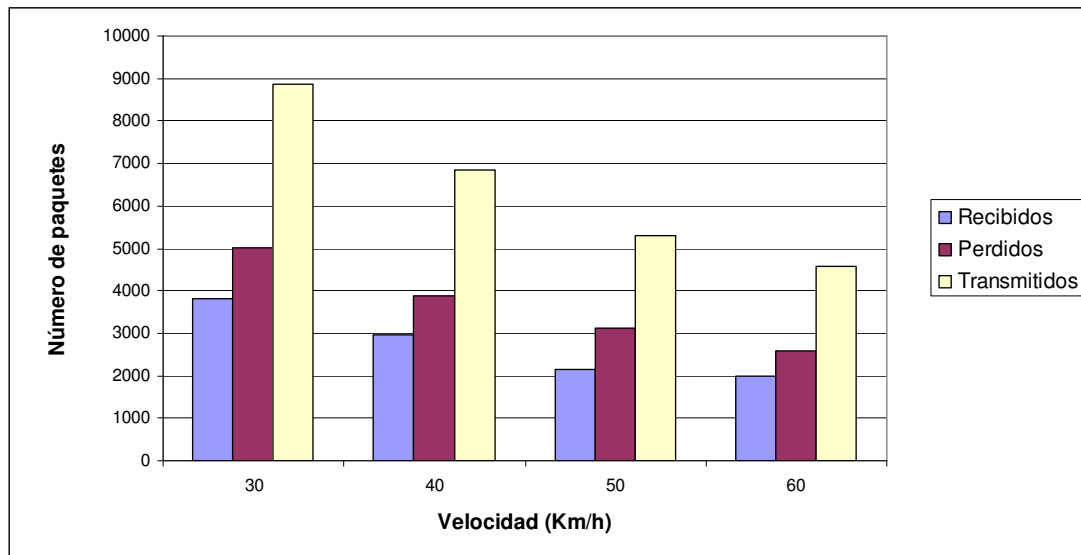


Figura 7.53. Vista del origen de la trayectoria recorrida por el coche desde la posición del receptor.



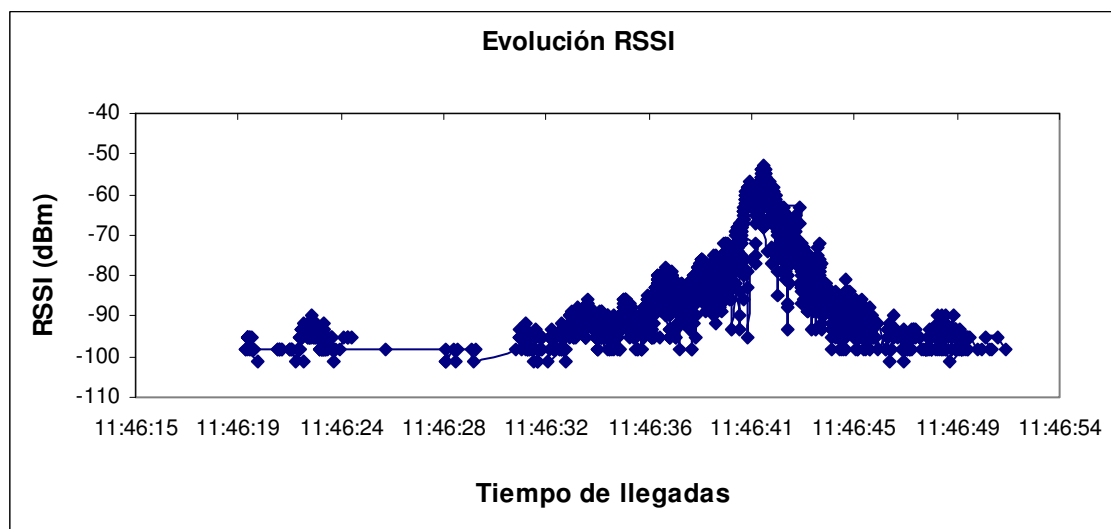
**Figura 7.54.** Vista del final de la trayectoria recorrida por el coche desde la posición del receptor.

Se realizaron cuatro pasadas con el coche, a velocidades de 30, 40, 50 y 60 Km/h cada una, tratando de mantener la velocidad del vehiculo lo mas constante posible. El la figura 7.55 se pueden ver los paquetes enviados, los recibidos y los perdidos en función de las diferentes velocidades del coche. El transmisor emitía continuamente, y a la misma tasa de transmisión. La reducción de los paquetes transmitidos se debe a que al aumentar la velocidad del coche, menor tiempo permanecía el receptor recibiendo paquetes.



**Figura 7.55.** Número de paquetes recibidos y perdidos en función de la velocidad del coche.

En la figura 7.56 está representada la evolución de RSSI de los paquetes recibidos en función del tiempo (por lo tanto de la distancia entre transmisor y receptor) para una velocidad de 60 Km/h.



**Figura 7.56.** Evolución temporal de RSSI de los paquetes recibidos.



Como era de esperar, la potencia de los paquetes recibidos va aumentando conforme el transmisor se acerca al receptor, alcanzando el máximo a la altura de este, y volviendo a disminuir al alejarse. Cabe destacar que en la parte inicial del trayecto llegan paquetes de una distancia mayor que de la parte final. Es posible que esto se deba a que existe un edificio en la parte inicial en el que pueden surgir reflexiones que hagan que la señal llegue hasta el receptor (ver figura 7.53).

Como trabajos futuros, sería de gran interés analizar la utilización de antenas de mejores características, así como la evolución del número de paquetes perdidos utilizando ACKs.



## 8.-Conclusiones

Las conclusiones del trabajo realizado han sido las siguientes:

- El software de trazado de rayos proporciona estimaciones de potencia recibida bastante buenas, permitiendo realizar análisis más completos de los escenarios.
- Debido a las características de propagación en entornos interiores complejos (propagación multitrayecto por difracción, reflexión, refracción y dispersión), se hace muy difícil predecir los lugares ideales para la correcta colocación de los dispositivos inalámbricos.
- En un entorno interior estático, el nivel de potencia recibida por los dispositivos inalámbricos depende fuertemente de la posición de los elementos receptores, lo que tiene un gran impacto en el funcionamiento de los sensores, no sólo en los límites de sensibilidad, también en la capacidad global del sistema, que depende del nivel de señal y de la relación señal-ruido (SNR).
- Al depender fuertemente la potencia recibida de la posición de los elementos de la red (pequeñas variaciones en la posición pueden acarrear grandes cambios en la potencia recibida), se hace necesaria una evaluación de los diferentes puntos donde se deberían colocar los sensores, para garantizar la calidad mínima exigida para la transmisión.
- La potencia recibida por los dispositivos inalámbricos está directamente relacionada con el PER.
- ZigBee es muy robusto con respecto a la pérdida de paquetes, ya que en una transmisión sin ACKs comienza a perderlos en cantidades importantes a niveles de potencia recibida cercanas a su límite de sensibilidad.
- La utilización de ACKs hace que los paquetes perdidos tienda a cero, a excepción de casos en el límite de sensibilidad. Es decir, la utilización de ACKs disminuye notablemente el PER.
- Además de la posición de los dispositivos, otros factores influyen fuertemente en la potencia recibida, como la calidad de la antena, la orientación de la antena y la cercanía de obstáculos.





- La presencia o tránsito de personas afecta considerablemente a las prestaciones de la comunicación, aumentando considerablemente el PER si la transmisión no dispone de los suficientes ACKs.
- Los resultados confirman la gran influencia del entorno, haciendo cada escenario totalmente diferente a los demás.
- En los dispositivos sin la capacidad de autorregular el nivel de potencia transmitida, será clave encontrar la posición adecuada en el escenario que haga mínimo el nivel de potencia que deba transmitir, ya que el consumo de corriente de los dispositivos depende principalmente del nivel de potencia al que transmiten.
- En los dispositivos con capacidad de autorregular el nivel de potencia de transmisión para que se mantenga un nivel de potencia constante en recepción, hay que tener en cuenta que al aumentar la distancia entre dispositivos, el nivel de potencia que deberán transmitir será mayor, por lo que también aumentará el consumo.
- La potencia recibida está relacionada directamente con la calidad del enlace (PER), y son parámetros indispensables para un correcto funcionamiento de la red. Pero debido a que los dispositivos están alimentados por baterías, el consumo es un parámetro clave que hay que tener en cuenta, ya que si se minimiza el consumo de energía, la duración de las baterías será mucho mayor, ahorrando dinero (no hay que comprar otras si son desechables, o si hay que recargarlas) y de igual forma minimizar el impacto medioambiental.
- Los escenarios dinámicos son mucho más complejos que los estáticos, y requieren de un estudio mucho mayor.
- Existen otras tecnologías y electrodomésticos, como los microondas, que pueden interferir en el correcto funcionamiento de una red de sensores inalámbricos.
- A modo de resumen, se puede decir que los resultados obtenidos muestran que la radiopropagación en escenarios interiores es compleja, donde el nivel de potencia de señal recibida depende fuertemente de la topología, y que también afecta a otros parámetros como el PER, la capacidad de la red y el consumo de corriente. También muestran que realizando una buena planificación de la topología y la morfología de red, así como de la elección de los dispositivos inalámbricos más adecuados, el



consumo de potencia y el rendimiento global del sistema pueden ser optimizados considerablemente.



## 9.-Líneas futuras

Debido a la naturaleza del proyecto, las líneas de trabajo a seguir son muchas, ya que los escenarios en los que realizar medidas son innumerables, y dependerán en gran medida del ingenio y ocurrencias del investigador.

Con respecto a las medidas realizadas en este trabajo, se pueden mejorar y añadir estos aspectos:

- Realizar mayor número de medidas en los escenarios, para dar más robustez a los datos obtenidos, ya que las condiciones de los entornos son bastante cambiantes.
- Realizar más simulaciones, para intentar abarcar todos los casos medidos empíricamente, y poder comparar los resultados.
- Adquirir dispositivos de otras empresas y realizar medidas con ellos para ver las diferencias entre dispositivos.
- Obtener datos de simulación mediante otro software diferente.
- Repetir las medidas para transmisiones con diferente cantidad de ACKs y potencias de transmisión.
- Realizar medidas de RSSI y PER en redes compuestas por un mayor número de dispositivos inalámbricos.
- Realizar medidas para comprobar cómo afectan a la transmisión las interferencias de las redes WiFi y del electrodoméstico microondas.
- Obtener más datos de escenarios dinámicos, como el realizado con un automóvil.

Existen otros escenarios en los que empezar a realizar medidas, como son:

- Medidas de movilidad con dispositivos instalados en el cuerpo humano, o en la ropa.
- Medidas con dispositivos inalámbricos en un entorno doméstico.
- Sería muy interesante realizar medidas en entornos mucho más complejos, como por ejemplo un hospital o un hotel.



## Bibliografía

- [1] Gislason, D., *ZigBee Wireless Networking*, Newnes. [www.newnespress.com](http://www.newnespress.com).
- [2] Farahani, S., *ZigBee Wireless Networks and Transceivers*, Newnes (2008), ISBN: 978-0-7506-8393-7.
- [3] Dargie, W., Poellabauer, C., *Fundamentals of Wireless Sensor Networks. Theory and Practice*, Wiley Series on Wireless Communications and Mobile Computing. Wiley (2010), ISBN: 978-0-470-99765-9.
- [4] Yang Xiao, Yi Pan, *Emerging Wireless LANs, Wireless PANs and Wireless MANs*, IEEE 802.11, IEEE 802.15, IEEE 802.16 Wireless Standard Family. Wiley (2009). ISBN: 978-0-471-72069-0.
- [5] Hernando Rábanos, J.M., *Transmisión por radio*, Editorial Centro de Estudios Ramón Areces, S.A. (1995). ISBN: 84-8004-146-3.
- [6] *IEEE Std 802.15.4™-2006*, IEEE (2006). ISBN: 0-7381-4997-7.
- [7] Manuales *ZigBee Alliance*, [www.zigbee.org](http://www.zigbee.org).
- [8] Manual Osciloscopio *Tektronix DPO 3014*.
- [9] Manual Analizador de espectros portátil *Agilent FieldFox N9912A*.
- [10] Manual Analizador de espectros *ADVANTEST R3131*.
- [11] Hoja de especificaciones de dispositivos *Xbee-Pro*, [www.digi.com](http://www.digi.com).
- [12] Hoja de especificaciones del sniffer de atmel *RZRAVEN*, [www.atmel.com](http://www.atmel.com).
- [13] Hojas de especificaciones de *Cyfi*, [www.cypress.com](http://www.cypress.com), [www.artaflex.com](http://www.artaflex.com).
- [14] Hoja de especificaciones de antena de *Cyfi*, [www.antenova.com](http://www.antenova.com).
- [15] Especificaciones de antena OAN-1070 de Level One, [www.level1.com](http://www.level1.com).
- [16] [www.meshnetics.com](http://www.meshnetics.com).
- [17] [www.atmel.com](http://www.atmel.com).
- [18] [www.lsr.com](http://www.lsr.com).



- [19] [www.cirronet.com](http://www.cirronet.com).
- [20] [www.panasonic.com](http://www.panasonic.com).
- [21] [www.digi.com](http://www.digi.com).
- [22] [www.st.com](http://www.st.com).
- [23] [www.oki.com](http://www.oki.com).



## ANEXOS

### A.- Congresos:

- [1]      **Autores:**      Peio Lopez, Juan Antonio Nazabal, Víctor Torres, Carlos Fernández-Valdivielso, Francisco Falcone.
- Título:**        Analysis of Topological impact in Wireless Indoor Sensor Networks.
- Publicación:** 2011 IEEE AP-S International Symposium on Antennas and Propagation and 2011 USNC/URSI National Radio Science Meeting.
- Aceptación:** Confirmada el 01/03/2011

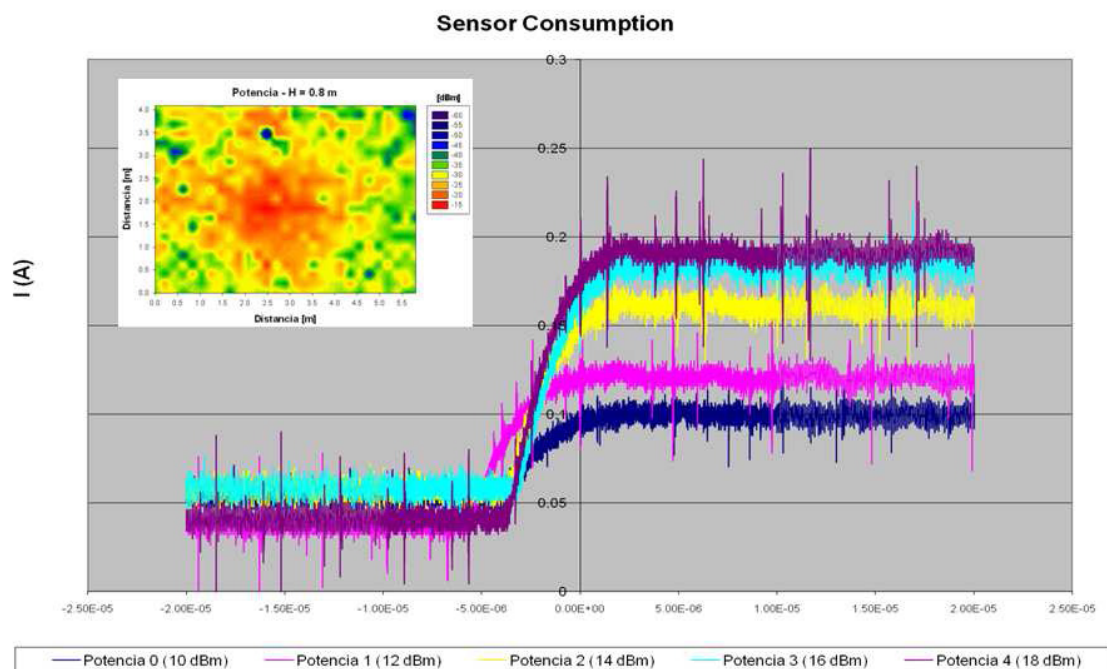


## Analysis of Topological impact in Wireless Indoor Sensor Networks

Peio López<sup>(1)</sup>, Juan Antonio Nazábal<sup>(1)</sup>, Víctor Torres<sup>(1)</sup>, Carlos Fernández<sup>(1)</sup> and (\*)F. Falcone<sup>(1)</sup>  
 (1) EE Dept, Public University of Navarra, Campus Arrosadía, E-31006, Pamplona, SPAIN,  
<http://www.unavarra.es>

The use of wireless sensor networks within domestic environments is leading towards the fast paced development of the so called smart homes. One of the key issues is to reduce energy consumption of the individual elements of these wireless sensor networks, due to the fact that in the near future, a great deal of these devices will be operating within a conventional indoor environment.

Radio channel features of indoor environments pose a challenge to energy consumption, due to the fact that the complexity of the scenario increases losses due to strong multipath propagation and multi-screen diffraction. In this paper, the topological influence of a layout of ZigBee sensors will be analyzed in terms of power consumption. Full 3D ray tracing simulation results are compared with current measurement values for different test cases. One of the results obtained is shown in fig 1, where different current curves have been measured for XBee sensor modules, within an indoor scenario (the inset in the figure corresponding to the ray tracing simulation results for the power distribution). These results aid in the future deployment and planning of complex indoor sensor networks, optimizing the overall power consumption without degrading system performance.





- [2]       **Autores:**       Peio Lopez, Juan Antonio Nazabal, Jesús Soret, Carlos Fernández-Valdivielso, Francisco Falcone.
- Título:**        Análisis del comportamiento de redes de sensores inalámbricas en entornos interiores heterogéneos.
- Publicación:** XXVI Simposio Nacional de la Unión Científica Internacional de Radio (URSI 2011).
- Aceptación:** Confirmada el 09/06/2011



# ANÁLISIS DEL COMPORTAMIENTO DE REDES DE SENSORES INALÁMBRICAS EN ENTORNOS INTERIORES HETEROGÉNEOS

Peio López<sup>1</sup>, Juan Antonio Nazabal<sup>1</sup>, Leire Azpilicueta<sup>1</sup>, Guillermo Martínez<sup>2</sup>, Raimundo García<sup>2</sup>, Jesús Soret<sup>2</sup>, José Torres<sup>2</sup>, Julio Martos<sup>2</sup>, Carlos Fernández-Valdivielso<sup>1</sup>, Francisco Falcone<sup>1</sup>.

peiolopeziturri@hotmail.com, juanantonio.nazabal@unavarra.es.

<sup>(1)</sup> Dpto IEE, Campus Arrosadia, UPNA; 31006, Pamplona, Navarra.

<sup>(2)</sup> Dpto. IEE, Universidad de Valencia, Burjassot, Valencia.

**Abstract-** This paper analyzes the impact of complex indoor environment in the deployment of a wireless sensor network infrastructure. The variable nature of the radio channel is analyzed by means of 3D ray launching simulation of an indoor scenario, in which wireless sensors typically used for environmental monitoring are placed. Measurement results indicate that adequate consideration of the network topology and morphology lead to optimal performance and power consumption reduction.

## I. INTRODUCCIÓN

El uso de redes de sensores inalámbricas está creciendo rápidamente en un gran número de campos de aplicación, como son la monitorización industrial, agricultura y ganadería, monitorización estructural, asistencia sanitaria, localización y seguimiento o seguridad y defensa, entre otros [1-8]. La tendencia en el futuro será el incremento del número de nodos con el fin de aumentar la interacción con los usuarios, así como obtener más información del entorno.

En el caso de sistemas de regulación automática, el escenario previsto es interior, con una alta complejidad relacionada con la topología así como con la morfología. Ello implica un incremento en las pérdidas debido a la absorción de los materiales y a la propagación multirrayecto. Los sistemas futuros tenderán a usar un gran número de sensores, en los cuales el consumo de energía y la capacidad serán parámetros clave a tener en cuenta.

En este trabajo se describirá la influencia de la topología y la morfología en escenarios interiores. Para ello, se realizarán simulaciones basadas en trazado de rayos 3D, que posteriormente serán comparadas con medidas sobre motas desarrolladas de manera específica para aplicaciones de monitorización de parámetros ambientales. Dichos resultados confirman la influencia del entorno, en particular en los valores de potencia consumida por los nodos.

## II. ANÁLISIS DEL ESCENARIO INTERIOR

A fin de analizar la influencia del escenario interior heterogéneo en las redes de sensores inalámbricas, a continuación se propone y se describe un escenario de prueba. Para ello, se han empleado un conjunto de motas desarrolladas con el fin de ser empleadas en aplicaciones de

monitorización y cuya configuración se describe a continuación.

El sistema consiste en varias motas. Cada una de ellas incluye elementos de sensor/actuador, un procesador PSoC, puertos de expansión y alimentación, y un transceptor CyFi. Dependiendo del rol que cada nodo adquiera en el protocolo, las motas podrán ser configuradas como maestro o como esclavo.

La topología básica es en estrella, en donde un nodo maestro administra un número concreto de nodos periféricos en modo esclavo. Cada mota contiene dos partes: una placa principal que contiene el microcontrolador, y una adicional que contiene la parte de radiofrecuencia. En la Fig.1 se muestra un prototipo del circuito de la mota empleada.



Fig. 1. Imagen del dispositivo.

La antena es el modelo Rufa 2.4 GHz SMD de Antenova, que trabaja en el rango 2.4-2.5 GHz. Su ganancia media es de -1.2dBi, siendo por lo tanto una antena de prestaciones moderadas, limitando el funcionamiento de la misma [9].

Con el objeto de obtener estimaciones de la influencia del entorno interior, se han obtenido resultados de simulación de planificación radioeléctrica. El análisis de radiopropagación se puede realizar mediante medidas de métodos empíricos, basados en aproximaciones estadísticas y técnicas de regresión no-lineales o basados en métodos deterministas (aproximación por óptica geométrica mediante trazado de rayos o lanzado de rayos) [10-14]. La ventaja del segundo tipo de técnica de simulación es su alta precisión a la vez que mantiene un nivel medio de complejidad computacional.



Con el fin de obtener las estimaciones de los cálculos de planificación radioeléctrica, se ha implementado un código de cálculo de lanzamiento de rayos 3D, realizado en Matlab. Se ha creado el escenario interior, teniendo en cuenta los parámetros de los materiales de todos los elementos presentes en el escenario (permitividad dieléctrica y la tangente de pérdidas) así como la topología del mismo (i.e., paredes, ventanas, muebles, etc.). Los fenómenos electromagnéticos de la reflexión, refracción y difracción de primer orden también han sido tomados en cuenta. El volumen del escenario de la simulación se divide en cuboides, con sus respectivos parámetros de simulación: frecuencia de operación, número de antenas transmisoras, número de rebotes y resolución angular del trazado de rayos. Una vez que las antenas transmisoras han sido definidas, se lanza la potencia en un ángulo sólido dado por el diagrama de radiación de las antenas. En Fig.4 se observa el lanzamiento de rayos para el trazado de rayos en un escenario interior definido.

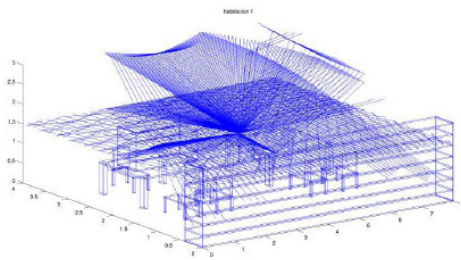


Fig. 2. Esquemático del trazado de rayos 3D en un escenario interior.

El escenario escogido para la simulación ha sido el laboratorio de Radiocomunicación de la Universidad Pública de Navarra. En él se incluyen multitud de elementos como mesas, sillas, armarios y una gran cantidad de instrumentos. La Fig.5 muestra el escenario analizado.



Fig. 3. Laboratorio de Radiocomunicación, UPNA.

Se han obtenido resultados de simulación de trazado de rayos 3D para todo el volumen del laboratorio. Fig.6 y Fig.7 representan el nivel de potencia recibida en los planos bidimensionales a 0.81m de altura (la altura de las mesas del laboratorio) y a 2.3m respectivamente. La antena transmisora

tiene la potencia y parámetros de transmisión equivalentes a un sistema ZigBee convencional.

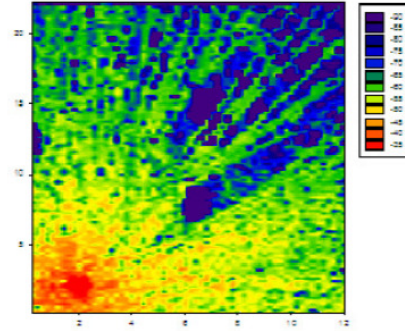


Fig. 4. Resultados de simulación (RSSI) del trazado de rayos 3D para un plano bidimensional a 0.81m de altura.

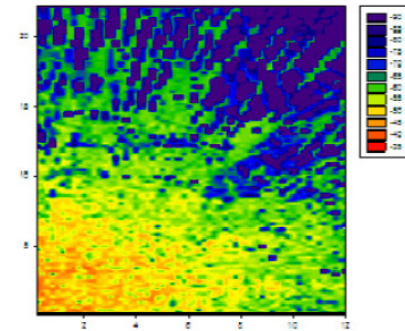


Fig. 5. Resultados de simulación (RSSI) del trazado de rayos 3D para un plano bidimensional a 2.3m de altura.

Como se puede observar en las dos figuras anteriores, el nivel de potencia recibida depende fuertemente de la posición de los elementos receptores. Las variaciones pueden ser del orden de 10 dB en 1 metro, lo que tiene un gran impacto en el funcionamiento de los sensores, no sólo en los límites de sensibilidad de los receptores, también en la capacidad global del sistema, que depende del nivel de señal y de la relación señal-ruido (SNR).

La figura 8 muestra los resultados de simulación, para las dos alturas previamente mostradas, a lo largo de una radial (eje x) en el escenario. La variación de la señal es debida al efecto de las componentes multitrayecto típicas en un entorno interior complejo.

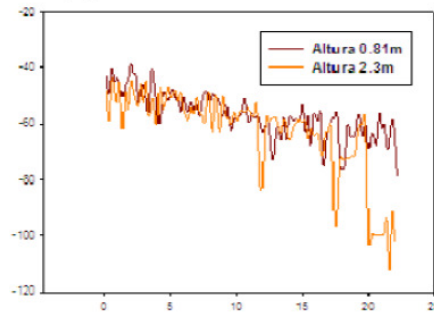


Fig. 6. Resultados de simulación para las alturas 0.81m y 2.3m, a lo largo del eje x del escenario interior analizado.

Una vez visto que la topología juega un papel clave en el funcionamiento global del sistema, a continuación se presentarán los resultados de las medidas. En la figura 9 se presenta el plano del escenario interior donde se han realizado las medidas. El laboratorio está dividido en dos zonas, separadas por varias estanterías metálicas.

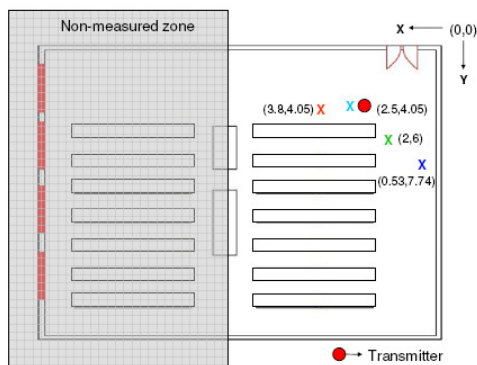


Fig. 7. Plano del escenario interior (Laboratorio de Radiocomunicación, Universidad Pública de Navarra).

Las medidas han sido realizadas estableciendo un elemento coordinador y un sensor inalámbrico. Las motas han sido programadas para proporcionar alta tasa de transmisión de paquetes, incrementando el nivel de la señal radio, y al mismo tiempo, el consumo de corriente. En Fig.10 se muestra el espectro de la señal de los dispositivos CyFi, medido en el escenario a analizar con un analizador de espectros en la coordenada (2,4).

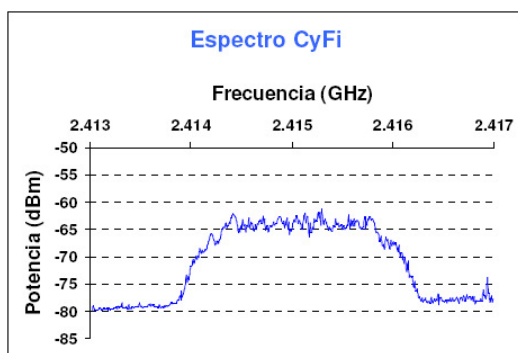


Fig. 8. Espectro de las motas CyFi en las coordenadas (2,4).

Han sido medidos los consumos de corriente de las motas en varias posiciones dentro del laboratorio, tanto en standby como transmitiendo. Los resultados se pueden observar en la figura 11 y la figura 12.

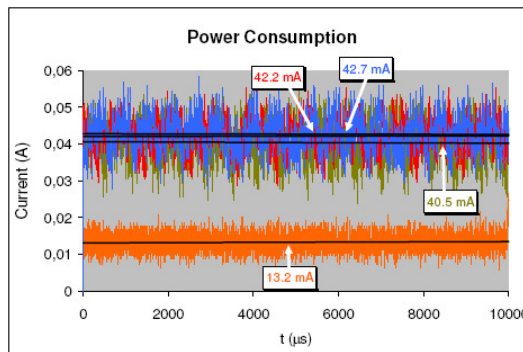


Fig. 9. Variación en el consumo de potencia de las motas en función de la posición. La curva inferior (13.2 mA) es para el modo en standby. El resto de las curvas van desde la posición más cercana a la más lejana.

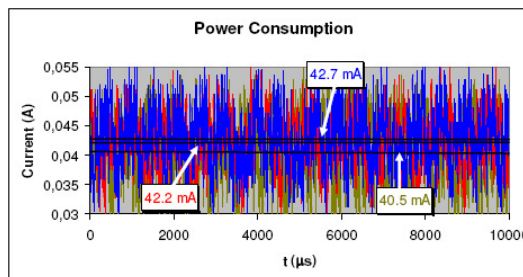


Fig. 10. Detalle del consumo de potencia de la mota en el escenario interior. Los valores medios de las corrientes son proporcionales a la separación entre motas.

Como puede apreciarse en la figura 10, existe una clara distinción entre el modo standby y el modo en transmisión, la cual era de esperar debido al normal procedimiento de operación en un transceptor inalámbrico, cuando la placa de RF está en modo activo o en modo idle.

En la figura 11, se presentan diferentes valores medios de corrientes, obtenidos para diferentes posiciones del sensor dentro del escenario interior. Al aumentar la distancia, el nivel de consumo de potencia también crece, como era de esperar debido al control automático de ganancia. Por tanto, al aumentar la distancia y (en principio) disminuir el nivel de potencia recibida, mayor es la demanda de potencia del receptor, teniendo un impacto directo en la eficiencia energética global de la red de sensores inalámbricos desplegada.

### III. CONCLUSIONES

En este trabajo se describe la influencia de la topología y la morfología en la operación de una red de sensores inalámbricos. Se ha analizado un escenario interior mediante trazado de rayos 3D y medidas con sensores inalámbricos. Los resultados muestran que la radiopropagación en escenarios interiores es compleja, donde el nivel de potencia de señal recibida depende fuertemente de la topología, que también afecta a otros parámetros como la capacidad de la red de sensores inalámbricos. Estos resultados muestran que considerando los condicionantes de planificación radioeléctrica en el uso de redes de sensores inalámbricos, el



consumo de potencia y el rendimiento global del sistema pueden ser optimizados considerablemente.

#### AGRADECIMIENTOS

Las labores de diseño y construcción de los elementos electrónicos han sido soportados por el proyecto INIA RTA-2007-00029-C02-02.

Este trabajo se ha financiado gracias al Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2008-2011, en concreto del proyecto Urbótica dentro del subprograma Movilidad Sostenible y el Cambio Modal en el Transporte.

Los resultados de simulación han sido obtenidos gracias a la financiación del proyecto IIM13185.RI1, Requisitos tecnológicos para actividad sanitaria rápida (FASTER), financiado por el Departamento de Innovación, Empresa y Empleo, Gobierno de Navarra

#### REFERENCIAS

- [1] Bose, R., *Sensor Networks – Motes, Smart Spaces, and Beyond*, IEEE Pervasive Computing, Vol. 8, Issue 3, pp. 84-90 (2009). ISSN: 1536-1268
- [2] Grosse, C.U., Glaser, S.D., Knuger, M., *Initial development of wireless acoustic emission sensor Motes for civil infrastructure state monitoring*, Smart Structures and Systems, Vol. 6, Issue 3, pp. 197-209 (2010)
- [3] Buckner, B.D., Markov, V., Lai, L.C., Earthman, J.C., *Laserscanning structural health monitoring with wireless sensor motes*, Optical Engineering, Vol. 47, Issue 5, Art. No. 054402 (2008), ISSN: 0091-3286
- [4] Berisha, V., Kwon, H., Spanias, A., *Real-time acoustic monitoring using wireless sensor motes*, Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 847-850 (2006). ISSN: 0277-674X
- [5] Trubilowicz, J., Cai, K., Weiler, M., *Viability of motes for hydrological measurements*, Water Resources Research, Vol. 45, Art. No. W00D22 (2009). ISSN: 0043-1397
- [6] Kuang, K.S.C., Quek, S.T., Maalej, M., *Remote flood monitoring system based on plastic optical fibres and wireless motes*, Sensors and Actuators A – Physical, Vol. 147, Issue 2, pp. 449-455 (2008). ISSN: 0924-4247
- [7] Yunseop, K., Evans, R.G., Iversen, W.M., *Remote sensing and control of an irrigation system using a distributed wireless sensor network*, IEEE Transactions on Instrumentation and Measurement, Vol. 57, Issue 7, pp. 1379-1387 (2008). ISSN: 0018-9456
- [8] Ruiz-Garcia, L., Barreiro, P., Robla, J.I., Lunadei, L., *Testing ZigBee motes for monitoring refrigerated vegetable transportation under real conditions*, Sensors, Vol. 10, Issue 5, pp. 4968-4982 (2010). ISSN: 1424-8220
- [9] ANTENOVA Rufa 2.4 GHz SMD, Part No. A5839
- [10] Magdy F. Iskander, Zhengqing Yun, *“Propagation Prediction Models for Wireless Communication Systems. IEEE Transaction on Microwave Theory and Techniques”*, Vol. 50, No. 3, March 2002
- [11] Danilo Erricolo and Piergiorgio L. E. Uslenghi, *Propagation Path Loss—A Comparison Between Ray-Tracing Approach and Empirical Models*, IEEE Transaction on Antennas and Propagation, Vol. 50m, No. 5, May 2002
- [12] M. Thiel, K. Sarabandi, *“3D-Wave Propagation Analysis of Indoor Wireless Channels Utilizing Hybrid Methods”*, IEEE Trans. Ant. Prop., Vol 57, No 5, pp 1539-1546, Mayo 2009
- [13] M. Tolstrup, *“Indoor Radio Planning”*, John Wiley, 2008, ISBN: 978-0-470-05769-8
- [14] J. Zhang, G. de la Roche, *“Femtocell. Technologies and Deployment”*, John Wiley, 2010, ISBN: 978-0-470-74298-3.

## B.- Código de los programas Java:

En este anexo se adjunta el código de los programas que se crearon en Java para la configuración y lectura de los datos de los dispositivos XBee-Pro y los CyFi de Valencia.

Conforme se iban realizando medidas en los distintos escenarios, los programas iban siendo actualizados y mejorados, añadiéndoles más funciones y corrigiendo errores. Los códigos aquí mostrados son los de las últimas versiones de cada programa.

El programa utilizado para la creación de estos programas fue el NetBeans IDE 7.0 (figura B.1).

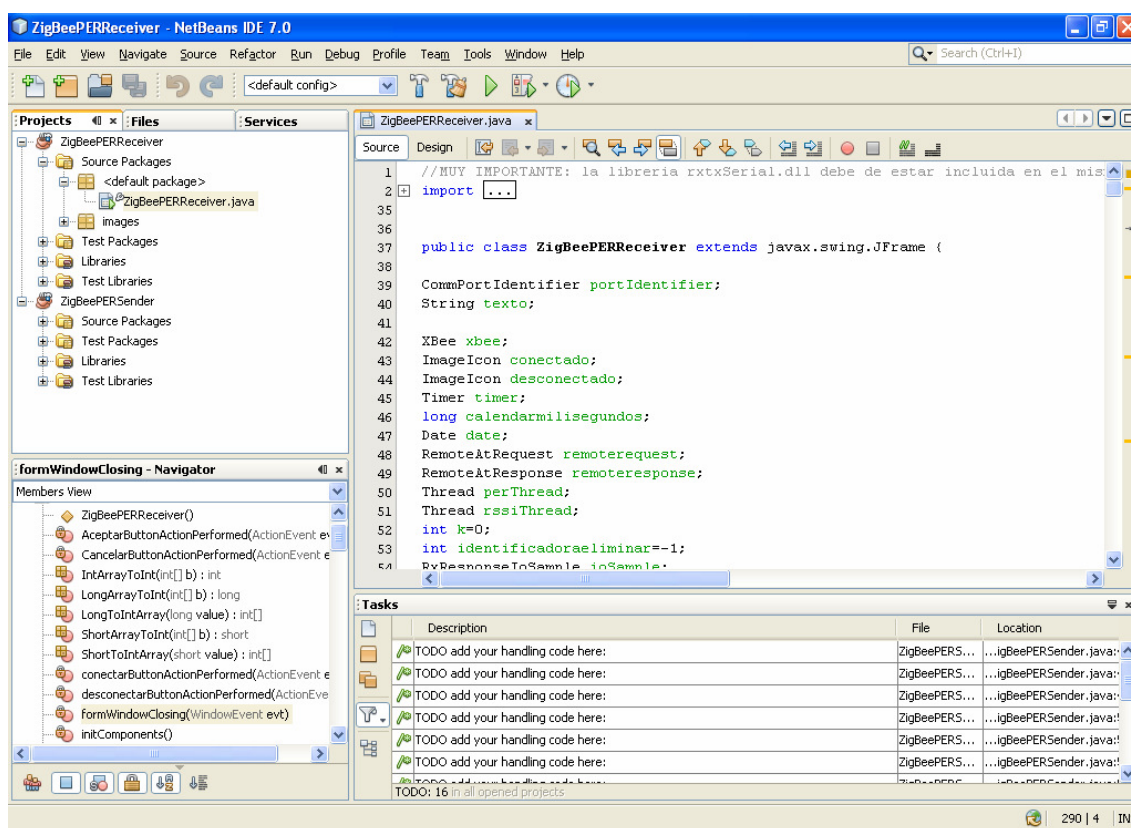


Figura B.1. Captura de pantalla del programa NetBeans 7.0.

Para trabajar con los dispositivos XBee-Pro, se crearon los programas ZigBeePERReceiver y ZigBeePERSender. ZigBeePERSender configura los parámetros de transmisión de los dispositivos mediante los valores que introduce el usuario, mientras ZigBeePERReceiver obtiene los datos de los paquetes recibidos, como el tiempo de llegada y la potencia recibida.



## B.1.- ZigBeePERSender:

//MUY IMPORTANTE: la libreria rxtxSerial.dll debe de estar incluida en el mismo path que el ejecutable

```
public class ZigBeePERSender extends javax.swing.JFrame {

    CommPortIdentifier portIdentifier;
    String texto;

    XBee xbee;
    ImageIcon conectado;
    ImageIcon desconectado;
    Timer timer;
    Calendar calendar;
    long calendarmilisegundos;
    Date date;
    RemoteAtRequest remoterequest;
    RemoteAtResponse remoterresponse;
    Thread perThread;
    Thread rssiThread;
    int k=0;
    int identificadoraeliminar=-1;
    RxResponseIoSample ioSample;
    double vref=3.3;
    int[] direccion;
    int rssi;
    double valor;
    double voltaje;
    long timestamp;
    int rssibuttonstate=-1;
    int perbuttonstate=-1;
    boolean rssialive=false;
    boolean peralive=false;

    public ZigBeePERSender()
    {
        this.conectado=new ImageIcon(getClass().getResource("/images/conectado.png"));
        this.desconectado=new ImageIcon(getClass().getResource("/images/desconectado.png"));
        rssialive=false;
        peralive=false;
        initComponents();

        rssibuttonstate=0;
        perbuttonstate=0;
        rssiButton.setBackground(Color.GRAY);
        perButton.setBackground(Color.GRAY);
        textField.setText("NO CONECTADO");

        /*
        remoterequest = new RemoteAtRequest(XBeeAddress64.BROADCAST, "DB"); //peticion
        broadcast de numero de serie high
        System.out.println("request:"+remoterequest.toString());
        */
    }
}
```



```

remoterequest = new RemoteAtRequest(XBeeRequest.DEFAULT_FRAME_ID,
XBeeAddress64.BROADCAST, XBeeAddress16.BROADCAST, false, "DB");
System.out.println("request:"+remoterequest.toString());
*/

remoterequest = new RemoteAtRequest(2, new XBeeAddress64
(0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff), new XBeeAddress16(0xff,0xfe), false, "DB");
System.out.println("request:"+remoterequest);
this.setLocationRelativeTo( null );
}

private void formWindowClosing(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    desconectarButtonActionPerformed(null);
}

private void CancelarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ConnectionDialog.dispose();
}

private void AceptarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    int timeout=0;
    int databits=SerialPort.DATABITS_8;
    //int stopbits=SerialPort.STOPBITS_1; //opcion a utilizar salvo 112500
    int stopbits=SerialPort.STOPBITS_2; //opcion a utilizar a 112500 para compensar el mal
funcionamiento del chip
    int paridad=SerialPort.PARITY_NONE;
    int flowcontrol=SerialPort.FLOWCONTROL_NONE;

    xbee = new XBee();
    System.out.println("Com:"+ComComboBox.getSelectedItem().toString());
    System.out.println("datarate:"+DataRateComboBox.getSelectedItem().toString());
    System.out.println("datarate int
:"+Integer.parseInt(DataRateComboBox.getSelectedItem().toString()));
    try
    {
        //xbee.open(ComComboBox.getSelectedItem().toString(),
Integer.parseInt(DataRateComboBox.getSelectedItem().toString()));
        //openSerialPort(java.lang.String port, java.lang.String appName, int timeout, int baudRate,
int dataBits, int stopBits, int parity, int flowControl)
        //this.openSerialPort(port, "XBee", 0, baudRate, SerialPort.DATABITS_8,
SerialPort.STOPBITS_1, SerialPort.PARITY_NONE, SerialPort.FLOWCONTROL_NONE);
        xbee.open(ComComboBox.getSelectedItem().toString(), "Xbee", timeout,
Integer.parseInt(DataRateComboBox.getSelectedItem().toString()),
databits, stopbits, paridad, flowcontrol);
        System.out.println("open correcto");
        //System.out.println("estado conexion"+xbee.isConnected());

        conectarButton.setEnabled(false);
        desconectarButton.setEnabled(true);

```



```

textField.setText("CONECTADO");
desconectarButton.update(desconectarButton.getGraphics());
conectarButton.update(conectarButton.getGraphics());
}
catch (com.rapplogic.xbee.api.XBeeException e)
{
System.out.println("error en el open:"+e.getMessage());
}

//perThread = new PERThread();
//perThread.start();

ConnectionDialog.dispose();

}

private void desconectarButtonActionPerformed(java.awt.event.ActionEvent evt) {

if (xbee != null) xbee.close();
desconectarButton.setEnabled(false);
conectarButton.setEnabled(true);

}

private void conectarButtonActionPerformed(java.awt.event.ActionEvent evt) {

int k=0;

java.util.Enumeration<CommPortIdentifier> portEnum =
CommPortIdentifier.getPortIdentifiers();
ComComboBox.removeAllItems();
while ( portEnum.hasMoreElements() )
{
k++;
portIdentifier = portEnum.nextElement();
if (portIdentifier.getPortType() == CommPortIdentifier.PORT_SERIAL)
{
ComComboBox.addItem(portIdentifier.getName());
}
}

ConnectionDialog.setLocationRelativeTo(this);
ConnectionDialog.setEnabled(true);
ConnectionDialog.setVisible(true);

}

private void poweroffActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
formWindowClosing(null);
System.exit(0);
}

```





```
private void rssiCancelarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    RSSIDialog.dispose();
}

private void rssiAceptarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    if (perbuttonstate == 0)
    {
        rssiThread = new RSSIThread();
        rssiThread.start();
        rssibuttonstate=1;
        rssialive=true;
        rssiButton.setBackground(Color.BLUE);
    }
    RSSIDialog.dispose();
}

private void PERCancelarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    PERDialog.dispose();
}

private void PERAceptarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (rssibuttonstate == 0)
    {
        perThread = new PERThread();
        perThread.start();
        perbuttonstate=1;
        peralive=true;
        perButton.setBackground(Color.BLUE);
    }
    PERDialog.dispose();
}

private void rssiButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (perbuttonstate == 0)
    {
        rssiThread = new RSSIThread();
        rssiThread.start();
        rssibuttonstate=1;
        rssialive=true;
        rssiButton.setBackground(Color.BLUE);
    }
}

private void perButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //System.out.println("tron:"+perbuttonstate);
    if (perbuttonstate == 0) {
```



```

    PERDialog.setLocationRelativeTo(this);
    PERDialog.setEnabled(true);
    PERDialog.setVisible(true);
} else if (perbuttonstate == 1) {
    if (perThread != null) peralive=false;
    perButton.setEnabled(false);
    //perbuttonstate=0;
    //perButton.setBackground(Color.GRAY);
}
}
}

```

```
class PERThread extends Thread
```

```

{
int numeropaquetes=-1;
int direccion=-1;
TxRequest64 request;
int i=0;
int conacks=0;
public void run()
{
    textField.setText("ENVIANDO(PER) ...");
    numeropaquetes=(int)(Math.pow(10,numpaquetesComboBox.getSelectedIndex()));
    xbee.clearResponseQueue();
    if (numpaquetesComboBox.getSelectedIndex() == 6) numeropaquetes=500000;
    if (numpaquetesComboBox.getSelectedIndex() == 7)
numeropaquetes=(int)(Math.pow(10,(numpaquetesComboBox.getSelectedIndex()-1)));
    textField.setText("ENVIANDO "+numeropaquetes+" PAQUETES ...");
    //request = new TxRequest16(new XBeeAddress16(0, direccion), new int[] {-1});
    //request = new TxRequest16(new XBeeAddress16(0, direccion),
69,com.rapplogic.xbee.api.wpan.TxRequestBase.Option.DISABLE_ACK,new int[] {-1});
    //request = new TxRequest16(new XBeeAddress16(0, direccion),
69,com.rapplogic.xbee.api.wpan.TxRequestBase.Option.DISABLE_ACK,new int[]
{numpaquetesComboBox.getSelectedIndex()});
    //try{xbee.sendRequest(request);} catch (java.io.IOException e){System.out.println("error
request -1:"+e.getMessage());}

    if (!lackCheckBox.isSelected())
    {
        conacks=0;
        while (peralive & (i<numeropaquetes) )
        {
            //try{Thread.sleep(1000);} catch (java.lang.InterruptedExcepcion e) {}
            //request = new TxRequest16(new XBeeAddress16(0, direccion),
69,com.rapplogic.xbee.api.wpan.TxRequestBase.Option.DISABLE_ACK,new int[]
{contador});
            textField.setText("ENVIANDO PAQUETE:"+(i+1)+" DE "+numeropaquetes+" SIN
ACKS");
            request = new TxRequest64(XBeeAddress64.BROADCAST,
69,com.rapplogic.xbee.api.wpan.TxRequestBase.Option.DISABLE_ACK,intToIntArray(i));
            //request = new TxRequest16(new XBeeAddress16(0, direccion),
69,com.rapplogic.xbee.api.wpan.TxRequestBase.Option.DISABLE_ACK,intToIntArray(i));
            //try{xbee.sendRequest(request);} catch (java.io.IOException e){System.out.println("error
request:"+e.getMessage());}

```



```

    try {xbee.sendAsynchronous(request);} catch (com.rapplogic.xbee.api.XBeeException
e){System.out.println("error request:"+e.getMessage());}
    i++;
    }
    }
    else
    {
    while (peralive & (i<numeropaquetes) )
    {
    conacks=1;
    //try {Thread.sleep(1000);} catch (java.lang.InterruptedExcepcion e) {}
    //request = new TxRequest16(new XBeeAddress16(0, direccion),
69,com.rapplogic.xbee.api.wpan.TxRequestBase.Option.DISABLE_ACK,new int[]
{contador});
    textField.setText("ENVIANDO PAQUETE:"+i+" DE "+numeropaquetes+" CON
ACKS");
    request = new TxRequest64(XBeeAddress64.BROADCAST, 69,intToIntArray(i));
    //try {xbee.sendRequest(request);} catch (java.io.IOException e){System.out.println("error
request:"+e.getMessage());}
    try {xbee.sendAsynchronous(request);} catch (com.rapplogic.xbee.api.XBeeException
e){System.out.println("error request:"+e.getMessage());}
    i++;
    }
    }

    i=0;
    for (i=0;i<1000;i++)
    {
    //el 65,66,15,0 es el numero correspondiente a 1000001, el numero de paquete de paro
request = new TxRequest64(XBeeAddress64.BROADCAST, 69,new int[]
{65,66,15,0,numpaquetesComboBox.getSelectedIndex(),conacks});
    try {xbee.sendAsynchronous(request);} catch (com.rapplogic.xbee.api.XBeeException
e){System.out.println("error request:"+e.getMessage());}
    //try {xbee.sendRequest(request);} catch (java.io.IOException e){System.out.println("error
request -1:"+e.getMessage());}
    }
}

textField.setText("PAQUETES ENVIADOS");
Toolkit.getDefaultToolkit().beep(); //hace un pitido
perbuttonstate=0;
perButton.setBackground(Color.GRAY);
perButton.setEnabled(true);
}

}

class RSSIThread extends Thread
{
int direccion=-1;
TxRequest64 request;
XBeeResponse response;

```



```

boolean applychanges=false;
public void run()
{
    textField.setText("ENVIANDO(RSSI) ...");
    direccion=Integer.parseInt(RSSIdireccionTextField.getText());
    xbee.clearResponseQueue();
    //while (!Thread.interrupted())
    while (rssialive)
    {
        //System.out.println(rssiThread.isInterrupted());
        //System.out.println("antes del sleep");
        try{Thread.sleep(500);} catch (java.lang.InterruptedExcepcion e) {}

        //request = new TxRequest64(XBeeAddress64.BROADCAST,new int[] {66,66,66,66});
        //try {xbee.sendAsynchronous(request);} catch (com.rapplogic.xbee.api.XBeeException
    e){System.out.println("error request:"+e.getMessage());}

        remoterequest = new RemoteAtRequest(XBeeAddress64.BROADCAST, "SL"); //peticion
broadcast de numero de serie high
        try
        {
            response=xbee.sendSynchronous(remoterequest,12000);
            System.out.println(response.toString());
        }
        catch (com.rapplogic.xbee.api.XBeeException e){System.out.println("error
request:"+e.getMessage());}
    }

    rssibuttonstate=0;
    rssiButton.setBackground(Color.GRAY);
    rssiButton.setEnabled(true);
}

}

int[] intToIntArray(int value) {
    return new int[] {
        (int)((byte)(value)),
        (int)((byte)(value >>> 8)),
        (int)((byte)(value >>> 16)),
        (int)((byte) (value >>> 24))};
}

int IntArrayToInt(int [] b) {
    return (b[3] << 24)
        + ((b[2] & 0xFF) << 16)
        + ((b[1] & 0xFF) << 8)
        + (b[0] & 0xFF);
}

```



```
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ZigBeePERSender().setVisible(true);
        }
    });
}
```



## B.2.- ZigBeePERReceiver:

//MUY IMPORTANTE: la libreria rxtxSerial.dll debe de estar incluida en el mismo path que el ejecutable

```
public class ZigBeePERReceiver extends javax.swing.JFrame {

    CommPortIdentifier portIdentifier;
    String texto;

    XBee xbee;
    ImageIcon conectado;
    ImageIcon desconectado;
    Timer timer;
    long calendarmilisegundos;
    Date date;
    RemoteAtRequest remoterequest;
    RemoteAtResponse remoterresponse;
    Thread perThread;
    Thread rssiThread;
    int k=0;
    int identificadoraeliminar=-1;
    RxResponseIoSample ioSample;
    int rssi;
    int rssibuttonstate=-1;
    int perbuttonstate=-1;
    boolean rssialive=false;
    boolean peralive=false;

    public ZigBeePERReceiver()
    {
        this.conectado=new ImageIcon(getClass().getResource("/images/conectado.png"));
        this.desconectado=new ImageIcon(getClass().getResource("/images/desconectado.png"));
        rssialive=false;
        peralive=false;

        initComponents();

        rssibuttonstate=0;
        perbuttonstate=0;
        rssiButton.setBackground(Color.GRAY);
        perButton.setBackground(Color.GRAY);

        textField.setText("NO CONECTADO");
        this.setLocationRelativeTo( null );
    }

    private void formWindowClosing(java.awt.event.WindowEvent evt) {
        // TODO add your handling code here:
        desconectarButtonActionPerformed(null);
    }
}
```



```

private void CancelarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ConnectionDialog.dispose();
}

private void AceptarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    int timeout=0;
    int databits=SerialPort.DATABITS_8;
    //int stopbits=SerialPort.STOPBITS_1; //opcion a utilizar salvo 112500
    int stopbits=SerialPort.STOPBITS_2; //opcion a utilizar a 112500 para compensar el mal
funcionamiento del chip
    int paridad=SerialPort.PARITY_NONE;
    int flowcontrol=SerialPort.FLOWCONTROL_NONE;

    xbee = new XBee();
    System.out.println("Com:"+ComComboBox.getSelectedItem().toString());
    System.out.println("datarate:"+DataRateComboBox.getSelectedItem().toString());
    System.out.println("datarate int
:"+Integer.parseInt(DataRateComboBox.getSelectedItem().toString()));
    try
    {
        //xbee.open(ComComboBox.getSelectedItem().toString(),
Integer.parseInt(DataRateComboBox.getSelectedItem().toString()));
        //openSerialPort(java.lang.String port, java.lang.String appName, int timeout, int baudRate,
int dataBits, int stopBits, int parity, int flowControl)
        //this.openSerialPort(port, "XBee", 0, baudRate, SerialPort.DATABITS_8,
SerialPort.STOPBITS_1, SerialPort.PARITY_NONE, SerialPort.FLOWCONTROL_NONE);
        xbee.open(ComComboBox.getSelectedItem().toString(), "Xbee", timeout,
Integer.parseInt(DataRateComboBox.getSelectedItem().toString()),
databits, stopbits, paridad, flowcontrol);
        System.out.println("open correcto");
        //System.out.println("estado conexion"+xbee.isConnected());
        conectarButton.setEnabled(false);
        desconectarButton.setEnabled(true);
        textField.setText("CONECTADO");
        desconectarButton.update(desconectarButton.getGraphics());
        conectarButton.update(conectarButton.getGraphics());
    }
    catch (com.rapplogic.xbee.api.XBeeException e)
    {
        System.out.println("error en el open:"+e.getMessage());
    }

    //perThread = new PERThread();
    //perThread.start();

    ConnectionDialog.dispose();

}

```



```

private void desconectarButtonActionPerformed(java.awt.event.ActionEvent evt) {

    if (xbee != null) xbee.close();
    desconectarButton.setEnabled(false);
    conectarButton.setEnabled(true);

}

private void conectarButtonActionPerformed(java.awt.event.ActionEvent evt) {

    int k=0;

    java.util.Enumeration<CommPortIdentifier> portEnum =
CommPortIdentifier.getPortIdentifiers();
    ComComboBox.removeAllItems();
    while ( portEnum.hasMoreElements() )
    {
        k++;
        portIdentifier = portEnum.nextElement();
        if (portIdentifier.getPortType() == CommPortIdentifier.PORT_SERIAL)
        {
            ComComboBox.addItem(portIdentifier.getName());
        }
    }

    ConnectionDialog.setLocationRelativeTo(this);
    ConnectionDialog.setEnabled(true);
    ConnectionDialog.setVisible(true);

}

private void poweroffActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    formWindowClosing(null);
    System.exit(0);
}

private void rssiButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (perbuttonstate == 0)
    {
        if (rssibuttonstate == 0)
        {
            rssiThread = new RSSIThread();
            rssiThread.start();
            rssibuttonstate=1;
            rssialive=true;
            rssiButton.setBackground(Color.BLUE);
        }
        else if (rssibuttonstate == 1)
        {
            if (rssiThread != null) rssialive=false;
            rssiButton.setEnabled(false);
        }
    }
}

```





```

        //rssibuttonstate=0;
        //rssiButton.setBackground(Color.GRAY);
    }
}

private void perButtonActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
    if (rssibuttonstate == 0)
    {
        if (perbuttonstate == 0)
        {
            perThread = new PERThread();
            perThread.start();
            perbuttonstate=1;
            peralive=true;
            perButton.setBackground(Color.BLUE);
        }
        else if (perbuttonstate == 1)
        {
            if (perThread != null) peralive=false;
            perButton.setEnabled(true);
            //perbuttonstate=0;
            //perButton.setBackground(Color.GRAY);
        }
    }
}

class PERThread extends Thread
{

    XBeeResponse response;
    byte[] bytearray;
    int numeropaquetes=-1;
    byte datorecibidoarray[]=new byte[4];
    int datorecibido=0;
    double rssi=0;
    int direccion;
    int i=0;
    Calendar calendar1,calendar2;
    Calendar llegadaC;
    long llegada;
    long timestamp1,timestamp2;
    double rssimedia=0;
    double per=-1;
    String cadena="";
    int conacks=0;
    BufferedWriter bufferedWriter;
    int numerodepaquete;
    int numerodepaqueta[]=new int[4];
    AtCommandResponse atResponse;
    RemoteAtResponse remoteresponse;

```



```

XBeeAddress64 remoteaddress64;
RxResponse64 rxResponse64;
int datos[];

public void run()
{
    textField.setText("ESCUCHANDO(PER) ...");
    rssimedia=0;
    xbee.clearResponseQueue();
    calendar1=Calendar.getInstance();
    timestamp1=calendar1.getTimeInMillis();

    if (outputfileCheckBox.isSelected())
    {
        try {bufferedWriter = new BufferedWriter(new FileWriter("salida.txt"));} catch
        (IOException e){System.out.println("write error");}
    }

    textField.setText("ESPERANDO PAQUETES...");
    i=0;
    while (peralive)
    {
        try
        {
            response = xbee.getResponse();
            llegadaC=Calendar.getInstance();
            llegada=llegadaC.getTimeInMillis();
            if (response.getApiId() == ApiId.RX_64_RESPONSE)
            {
                rxResponse64=(RxResponse64)response;
                rssi=rxResponse64.getRssi();
                datos=rxResponse64.getData();
                remoteaddress64=rxResponse64.getRemoteAddress();
                //numerodepaquetea[0]=response.getProcessedPacketBytes()[7];
                //numerodepaquetea[1]=response.getProcessedPacketBytes()[8];
                //numerodepaquetea[2]=response.getProcessedPacketBytes()[9];
                //numerodepaquetea[3]=response.getProcessedPacketBytes()[10];
                //numerodepaquete=IntArrayToInt(numerodepaquetea);

                numerodepaquetea[0]=datos[0];
                numerodepaquetea[1]=datos[1];
                numerodepaquetea[2]=datos[2];
                numerodepaquetea[3]=datos[3];
                numerodepaquete=IntArrayToInt(datos);
                textField.setText("RECIBIENDO PAQUETE NUMERO:"+(i+1)+" CON NUMERO DE
                PAQUETE:"+numerodepaquete);
                //System.out.println("frame id"+response.getProcessedPacketBytes()[4]);
                //System.out.println("rssi:"+rssi+", paquete:"+numerodepaquete);
                //System.out.println("a[3]:"+numerodepaquetea[3]+", a[2]:"+numerodepaquetea[2]+",
                a[1]:"+numerodepaquetea[1]+", a[0]:"+numerodepaquetea[0]);
                rssi=rxResponse64.getRssi();
                //System.out.println("rssi"+rssi);
                rssimedia=rssimedia+rssi;
            }
        }
    }
}

```



```

if (outputfileCheckBox.isSelected())
{
try {bufferedWriter.write(""+llegada+";"+numerodepaquete+";"+rssi+";");}
catch (Exception e)
{
System.out.println("error escribiendo filas");
try {bufferedWriter.write("error:"+e.getLocalizedMessage()+" , error:"+e.getMessage());}
catch (Exception eo) {}
textField.setText("error:"+e.getLocalizedMessage()+" , error:"+e.getMessage());
}
try {bufferedWriter.newLine();} catch (Exception e) {System.out.println("error escribiendo
filas");}
}

if (numerodepaquete == 1000001)
{
datorecibido=datos[4];
conacks=datos[5];
numeropaquetes=(int)(Math.pow(10,datorecibido));
if (datorecibido == 6) numeropaquetes=500000;
if (datorecibido == 7) numeropaquetes=(int)(Math.pow(10,(datorecibido-1)));
if (conacks==0) cadena="SIN ACKS" ;
else if (conacks==1) cadena="CON ACKS" ;
break;
}
else
{
i++;
}
//System.out.println("dato recibido"+datorecibido);
//System.out.println("dato recibido"+response.toString());
}
}
catch (com.rapplogic.xbee.api.XBeeException e) {}
}

if (outputfileCheckBox.isSelected())
{
try {bufferedWriter.close();} catch (IOException e) {System.out.println("write error");}
}

calendar2=Calendar.getInstance();
timestamp2=calendar2.getTimeInMillis();
rssimedia=(rssimedia/((double)(i)));
per=(((double)(numeropaquetes))-((double)(i)))*100/((double)(numeropaquetes));
//System.out.println("paquetes enviados"+numeropaquetes);
//System.out.println("paquetes recibidos"+i);
//System.out.println("pper"+per);
textField.setText(""+cadena+"") PAQUETES RECIBIDOS "+i+"DE "+numeropaquetes+"
CON RSSI MEDIO "+rssimedia+" EN "+(timestamp2-timestamp1)+" MILISEGUNDOS,
PER:"+per);
perbuttonstate=0;
perButton.setBackground(Color.GRAY);

```



```

    perButton.setEnabled(true);
    }

}

class RSSIThread extends Thread
{

    XBeeResponse response;
    int rssi=0;
    int i=0;
    XBeeAddress64 remoteaddress64;
    RxResponse64 rxResponse64;
    public void run()
    {
        textField.setText("ESCUCHANDO(RSSI) ...");
        xbee.clearResponseQueue();
        i=0;
        while (rssialive)
        {
            //System.out.println(rssiThread.isInterrupted());
            try
            {
                //response = xbee.getResponse();
                //System.out.println("antes del getresponse");
                response = xbee.getResponse(3000);
                //System.out.println("respuesta:"+response.toString());
                if (response.getApiId() == ApiId.RX_64_RESPONSE)
                {
                    rxResponse64=(RxResponse64)response;
                    //System.out.println("recibida response:"+response.toString()+"con
id:"+response.getApiId());
                    rssi=rxResponse64.getRssi();
                    remoteaddress64=rxResponse64.getRemoteAddress();
                    i++;
                    if (i > 256) i=0;
                    //System.out.println("despues del getresponse");
                    //System.out.println("despues del rssi");
                    textField.setText("RSSI "+rssi+ "("+i+") DE LA
DIRECCION:"+remoteaddress64.toString());
                }
            }
            catch (com.rapplogic.xbee.api.XBeeException e)
            {
                System.out.println("timeout ocurrido");
            }
        }
        rssibuttonstate=0;
        rssiButton.setBackground(Color.GRAY);
        rssiButton.setEnabled(true);
    }

}

```



```
int[] intToArray(int value) {
    return new int[] {
        (int)((byte)(value)),
        (int)((byte)(value >>> 8)),
        (int)((byte)(value >>> 16)),
        (int)((byte)(value >>> 24));
    }
}
```

```
int IntArrayToInt(int [] b) {
    return (b[3] << 24
        + ((b[2] & 0xFF) << 16)
        + ((b[1] & 0xFF) << 8)
        + (b[0] & 0xFF));
}
```

```
int[] LongToArray(long value) {
    int [] b = new int[8];
    for(int i= 0; i < 8; i++)
    {
        b[7 - i] = (int)(value >>> (i * 8));
    }
    return b;
}
```

```
int[] ShortToArray(short value) {
    int [] b = new int[2];
    for(int i= 0; i < 2; i++)
    {
        b[1 - i] = (int)(value >>> (i * 8));
    }
    return b;
}
```

```
long LongArrayToInt(int [] b) {
    long ret=0;
    for(int i=0; i < 8; i++)
    {
        ret <<= 8;
        ret ^= (long)b[i] & 0xFF;
    }
    return ret;
}
```

```
short ShortArrayToInt(int [] b) {
    short ret=0;
    for(int i=0; i < 2; i++)
    {
        ret <<= 8;
        ret ^= (short)b[i] & 0xFF;
    }
}
```



```
return ret;
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ZigBeePERReceiver().setVisible(true);
        }
    });
}
}
```



### B.3.- ZigBeePERReceiverV:

Este es el programa que se creó para la obtención de los datos recibidos en los dispositivos que fueron cedidos por la Universidad de Valencia:

//MUY IMPORTANTE: la libreria rxtxSerial.dll debe de estar incluida en el mismo path que el ejecutable

```
public class ZigBeePERReceiverV extends javax.swing.JFrame implements
SerialPortEventListener {

CommPortIdentifier portIdentifier;
String texto;

ImageIcon conectado;
ImageIcon desconectado;
Timer timer;
long calendarmilisegundos;
Date date;
int identificadoraeliminar=-1;
CommPortIdentifier portId;
SerialPort serialPort=null;
//int selectedBaudRate = 0x1c200;
int selectedBaudRate = 115200;
int selectedBits = 8;
int selectedStop = 1;
int selectedParity = 0;
int selectedFlow = 0;
InputStream in;
ArrayList<Double> doublearray = new ArrayList<Double>();
int rssibuttonstate=-1;
int perbuttonstate=-1;
boolean rssialive=false;
boolean peralive=false;
int rssi;
int powerlevel;
double powerlevelmedio=0;
double rssimedia=0;
int
secuenciaanterior,secuenciaactual,secuenciainicial,secuenciafinal,maximossecuenciafinal,numer
odepaquetes,numerodellegadas,contadormaximossecuencia;
int maximossecuencia=256;
double per;
BufferedWriter bufferedWriter;
Calendar llegadaC;
long llegada=-1;
long llegadainicial=-1;
double rssimedia;
int i=0;
int media;
```



```

public ZigBeePERReceiverV()
{
    this.conectado=new ImageIcon(getClass().getResource("/images/conectado.png"));
    this.desconectado=new ImageIcon(getClass().getResource("/images/desconectado.png"));
    rssialive=false;
    peralive=false;

    initComponents();

    rssibuttonstate=0;
    perbuttonstate=0;
    rssiButton.setBackground(Color.GRAY);
    perButton.setBackground(Color.GRAY);

    textField.setText("NO CONECTADO");
    this.setLocationRelativeTo( null );
}

public void serialEvent(SerialPortEvent event) {
    switch (event.getEventType()) {
        case SerialPortEvent.BI:
        case SerialPortEvent.OE:
        case SerialPortEvent.FE:
        case SerialPortEvent.PE:
        case SerialPortEvent.CD:
        case SerialPortEvent.CTS:
        case SerialPortEvent.DSR:
        case SerialPortEvent.RI:
        case SerialPortEvent.OUTPUT_BUFFER_EMPTY:
            break;
        case SerialPortEvent.DATA_AVAILABLE:
            //System.out.println("data available");
            byte[] readBuffer = new byte[12];
            int numBytes=-1;
            try
            {
                while (in.available() > 0) numBytes = in.read(readBuffer);
                // System.out.println("bytes leidos"+((short)numBytes));
                //System.out.println("0:"+(short)readBuffer[0]);
                //System.out.println("1:"+(short)readBuffer[1]);
                //System.out.println("ret:"+(short)readBuffer[2]);
                //System.out.println("rssi:"+(short)readBuffer[3]);
                //System.out.println("secuencia:"+(short)readBuffer[8]);
                //System.out.println("nivel:"+(short)readBuffer[9]);
                rssi=(short)readBuffer[3];
                secuenciaactual=(short)readBuffer[8];
                powerlevel=(short)readBuffer[9];
                llegadaC=Calendar.getInstance();
                llegada=llegadaC.getTimeInMillis();
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
    }
}

```





```

if (rssialive == true)
{
media=Integer.parseInt(mediaTextField.getText());
//System.out.println("media:"+media);
if (media==0)
{
textField.setText("RSSI "+rssi+ "("+secuenciaactual+") power level:"+powerlevel);
}
else
{
if (llegadainicial== -1)
{
llegadainicial=llegada;
}
else
{
if ((llegada - llegadainicial) > media)
{
rssimedio=rssimedio/((double)i);
powerlevelmedio=powerlevelmedio/((double)i);
//System.out.println("tron:RSSI "+rssimedio+ ", power level:"+powerlevelmedio);
textField.setText("("+i+"): RSSI medio"+rssimedio+ ", power level
medio:"+powerlevelmedio);
i=0;
rssimedio=0;
powerlevelmedio=0;
llegadainicial=llegada;
}
else
{
rssimedio=rssimedio+rssi;
powerlevelmedio=powerlevelmedio+powerlevel;
i++;
//System.out.println("aki:RSSI "+rssimedio+ ", power level:"+powerlevelmedio+" i"+i);
}
}
}
}
else if (peralive == true)
{

if (secuenciainicial == -1)
{
//inicializamos
if (outputfileCheckBox.isSelected())
{
try{bufferedWriter = new BufferedWriter(new FileWriter("salida.txt"));} catch
(IOException e){System.out.println("write error");}
}
secuenciainicial=secuenciaactual;

```



```

        maximossecuenciafinal=(secuenciainicial+numerodepaquetes) / maximossecuencia; //el
resto de la division
        secuenciafinal=(secuenciainicial+numerodepaquetes) % maximossecuencia; //el modulo de
la division
        numerodellegadas=0;
        contadormaximossecuencia=0;
        secuencianterior=secuenciainicial;
        rssimedia=0;
    }
    else
    {
        if ((contadormaximossecuencia > maximossecuenciafinal) | ((secuenciaactual >
secuenciafinal ) & (contadormaximossecuencia == maximossecuenciafinal)))
        {
            // fin del per
            if (outputfileCheckBox.isSelected())
            {
                try {bufferedWriter.write("contadormaximossecuencia:"+contadormaximossecuencia+";maximo
ssecuenciafinal:"+maximossecuenciafinal+";secuenciaactual:"+secuenciaactual+";");} catch
(Exception e){System.out.println("error escribiendo filas");}
                try {bufferedWriter.newLine();} catch (Exception e){System.out.println("error
escribiendo filas");}
                try {bufferedWriter.close();} catch (IOException e){System.out.println("write error");}
            }
            per((((double)(numerodepaquetes))-
((double)(numerodellegadas)))*100)/((double)(numerodepaquetes));
            rssimedia=rssimedia/((double)numerodellegadas);
            textField.setText("PAQUETES RECIBIDOS "+numerodellegadas+"DE
"+numerodepaquetes+" CON RSSI MEDIO "+rssimedia+", PER:"+per);
            perbuttonstate=0;
            secuenciainicial=-1;
            peralive=false;
        }
        else
        {
            textField.setText("RECIBIENDO PAQUETE CON NUMERO DE
PAQUETE:"+secuenciaactual);
            numerodellegadas++;
            rssimedia=rssimedia+rssi;
            if (outputfileCheckBox.isSelected())
            {
                try {bufferedWriter.write(""+llegada+";"+secuenciaactual+";"+rssi+";"+powerlevel+";");} catch
(Exception e){System.out.println("error escribiendo filas");}
                try {bufferedWriter.newLine();} catch (Exception e){System.out.println("error
escribiendo filas");}
            }
            if (!(secuenciaactual > secuencianterior)) contadormaximossecuencia++;
        }
    }
}

```



```

    }
    }
    catch (IOException e) {System.out.println("error:"+e.getMessage());}
    break;
    }
}

private void formWindowClosing(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    desconectarButtonActionPerformed(null);
}

private void CancelarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ConnectionDialog.dispose();
}

private void AceptarButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    try
    {
        portId =
CommPortIdentifier.getPortIdentifier(ComComboBox.getSelectedItem().toString());
        serialPort = (SerialPort)portId.open(getClass().getName(), 2000);
        serialPort.setSerialPortParams(selectedBaudRate, selectedBits, selectedStop,
selectedParity);
        serialPort.setFlowControlMode(selectedFlow);
        serialPort.notifyOnDataAvailable(true);
        //serialPort.notifyOnFramingError(true);
        //serialPort.notifyOnParityError(true);
        in = serialPort.getInputStream();
        conectarButton.setEnabled(false);
        desconectarButton.setEnabled(true);
        textField.setText("CONECTADO");
        serialPort.addEventListener(this);
    }
    catch(Exception e)
    {

    }

    //perThread = new PERThread();
    //perThread.start();

    ConnectionDialog.dispose();

}

private void desconectarButtonActionPerformed(java.awt.event.ActionEvent evt) {

```



```

try{serialPort.close();} catch (Exception e){}
desconectarButton.setEnabled(false);
conectarButton.setEnabled(true);

}

private void conectarButtonActionPerformed(java.awt.event.ActionEvent evt) {

int k=0;

java.util.Enumeration<CommPortIdentifier> portEnum =
CommPortIdentifier.getPortIdentifiers();
ComComboBox.removeAllItems();
while ( portEnum.hasMoreElements() )
{
k++;
portIdentifier = portEnum.nextElement();
if (portIdentifier.getPortType() == CommPortIdentifier.PORT_SERIAL)
{
ComComboBox.addItem(portIdentifier.getName());
}
}

ConnectionDialog.setLocationRelativeTo(this);
ConnectionDialog.setEnabled(true);
ConnectionDialog.setVisible(true);

}

private void poweroffActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
formWindowClosing(null);
System.exit(0);
}

private void rssiButtonActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
if (perbuttonstate == 0)
{
if (rssibuttonstate == 0)
{
rssibuttonstate=1;
rssialive=true;
rssiButton.setBackground(Color.BLUE);
}
else if (rssibuttonstate == 1)
{
rssialive=false;
rssiButton.setEnabled(false);
//rssibuttonstate=0;
//rssiButton.setBackground(Color.GRAY);
}
}
}

```



```

    }

    private void perButtonActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:
        if (rssibuttonstate == 0)
        {
            if (perbuttonstate == 0)
            {

                PERDialog.setLocationRelativeTo(this);
                PERDialog.setEnabled(true);
                PERDialog.setVisible(true);

            }
            else if (perbuttonstate == 1)
            {
                peralive=false;
                perButton.setEnabled(true);
                //perbuttonstate=0;
                //perButton.setBackground(Color.GRAY);
            }
        }
    }

    private void PERCancelarButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        PERDialog.dispose();
    }

    private void PERAceptarButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (rssibuttonstate == 0) {
            numerodepaquetes=Integer.parseInt(numeropaquetesTextField.getText());
            secuenciainicial=-1;
            secuenciacfinal=-1;
            numerodellegadas=0;
            per=-1;
            perbuttonstate=1;
            peralive=true;
            perButton.setBackground(Color.BLUE);
        }
        PERDialog.dispose();
    }

    int[] intToIntArray(int value) {
        return new int[] {
            (int)((byte)(value)),
            (int)((byte)(value >>> 8)),
            (int)((byte)(value >>> 16)),
            (int)((byte)(value >>> 24))};
    }

```



```
int IntArrayToInt(int [] b) {
    return (b[3] << 24)
        + ((b[2] & 0xFF) << 16)
        + ((b[1] & 0xFF) << 8)
        + (b[0] & 0xFF);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new ZigBeePERReceiverV().setVisible(true);
        }
    });
}
```



## B.4.- Gráficas (Evolución temporal):

En el escenario número 6 fue necesario crear un programa para poder analizar la evolución temporal de los paquetes recibidos. Al programa se le llamó 'Graficas'. A continuación se muestra el código del programa que se creó:

```
public class Graficas extends javax.swing.JFrame {

    XYSeries xyserie[];
    XYSeriesCollection xydataset,xydatasetrss, xydatasetpaquetes;
    JFreeChart xychart;
    BufferedReader bufferedReader;
    BufferedWriter bufferedWriter;
    String readline;
    String splittedreadline[];
    Dimension screenSize;
    XYSeries seriesrss;
    XYSeries seriepaquetes;

    public Graficas()
    {

        initComponents();

        screenSize = Toolkit.getDefaultToolkit().getScreenSize();

        String timestampstring;
        String secuencia,secuenciaanterior,srss;
        long timestampactual,timestampanterior;
        long diferencia=0;
        String fechaexcel;
        Calendar calendar;
        NumberFormat doscifras = new DecimalFormat ( "00" ) ; // para mostrar unicamente dos cifras
        boolean llegatarde=false;

        int j=20;
        timestampanterior=-1;
        secuenciaanterior=null;

        try{bufferedReader = new BufferedReader(new FileReader("movilidad zigbee 30-60(original).txt"));} catch (IOException e){System.exit(-1);}
        try{bufferedWriter = new BufferedWriter(new FileWriter("movilidad zigbee "+j+"(excel).txt"));} catch (IOException e){System.exit(-1);}

        //parseamos los parametros del fichero de configuracion
        try
```



```

{

calendar=Calendar.getInstance();
while((readline=bufferedReader.readLine()) != null)
{
splittedreadline=readline.split(";");
timestampstring=splittedreadline[0];
secuencia=splittedreadline[1];
srssi=splittedreadline[2];

timestampactual=Long.parseLong(timestampstring);
if (timestampanterior == -1)
{
timestampanterior=timestampactual;
secuenciaanterior=secuencia;
}
if (!(Integer.parseInt(secuencia) > Integer.parseInt(secuenciaanterior)))
{
System.out.println("llegada tarde");
llegatarde=true;
}
}
if ((timestampactual - timestampanterior) > 30000)
{
bufferedWriter.close();
j=j+10;
bufferedWriter = new BufferedWriter(new FileWriter("movilidad zigbee "+j+" (excel).txt"));
}

calendar.setTimeInMillis(timestampactual);
fechaexcel=doscifras.format(calendar.get(Calendar.DATE))+"-
"+doscifras.format(calendar.get(Calendar.MONTH)+1)+
"+"+calendar.get(Calendar.YEAR)+"
"+doscifras.format(calendar.get(Calendar.HOUR_OF_DAY))+
":""+doscifras.format(calendar.get(Calendar.MINUTE))+":""+doscifras.format(calendar.get(Cale
ndar.SECOND))+".""+doscifras.format(calendar.get(Calendar.MILLISECOND));

if (llegatarde)
{
System.out.println("fecha:"+fechaexcel+ "j:"+j);
llegatarde=false;
}

bufferedWriter.write(timestampstring+";"+fechaexcel+";"+secuencia+";"+srssi);
bufferedWriter.newLine();

timestampanterior=timestampactual;

}
bufferedReader.close();
bufferedWriter.close();
}

```





```

catch (IOException e) {}

/*
try { bufferedReader = new BufferedReader(new FileReader("movilidad zigbee 30-60.txt")); }
catch (IOException e) { System.exit(-1); }
try { bufferedWriter = new BufferedWriter(new FileWriter("movilidad zigbee 30-60(excel).txt")); } catch (IOException e) { System.exit(-1); }

//parseamos los parametros del fichero de configuracion
try
{

calendar=Calendar.getInstance();
while((readline=bufferedReader.readLine()) != null)
{
splittedreadline=readline.split(";");
timestampstring=splittedreadline[0];
secuencia=splittedreadline[1];
srssi=splittedreadline[2];

timestamp=Long.parseLong(timestampstring);

calendar.setTimeInMillis(timestamp);
fechaexcel=doscifras.format(calendar.get(Calendar.DATE))+"-
"+doscifras.format(calendar.get(Calendar.MONTH)+1)+
"+"+calendar.get(Calendar.YEAR)+"
"+doscifras.format(calendar.get(Calendar.HOUR_OF_DAY))+

"."+doscifras.format(calendar.get(Calendar.MINUTE))+":"+"+doscifras.format(calendar.get(Cale
ndar.SECOND))+"."+doscifras.format(calendar.get(Calendar.MILLISECOND));

//System.out.println("fecha:"+fechaexcel);

bufferedWriter.write(timestampstring+";"+"+fechaexcel+";"+"+secuencia+";"+"+srssi);
bufferedWriter.newLine();
}
bufferedReader.close();
bufferedWriter.close();
}
catch (IOException e) {}
*/

/*
try
{
File f;
f=new File("consumos.csv");
if (!f.exists()) f.createNewFile();

```



```

bufferedReader = new BufferedReader(new FileReader(f));
}
catch (IOException o){}

try
{
int i=0;
xyserie = new XYSeries[5];
for (int j=0;j<5 ;j++)
{
xyserie[j] = new XYSeries("series "+j);
}
while((readline=bufferedReader.readLine()) != null)
{
splittedreadline=readline.split(";");
for (int j=0;j<splittedreadline.length-1 ;j++)
{
//xyserie[j].add(i,Double.parseDouble(splittedreadline[j+1]));

xyserie[j].add(Double.parseDouble(splittedreadline[0]),Double.parseDouble(splittedreadline[j+
1]));
}
i++;
}
bufferedReader.close();
}
catch (IOException e){System.out.println("tron"+e.getMessage());}

xydataset = new XYSeriesCollection();
for (int j=0;j<5 ;j++)
{
xydataset.addSeries(xyserie[j]);
}

xychart = ChartFactory.createXYLineChart(
    "consumos XBee", // chart title
    "Tiempo (s)", // domain axis label
    "Intensidad (A)", // range axis label
    xydataset, // data
    PlotOrientation.VERTICAL,
    true, // include legend
    true,
    false);

mainLabel.setSize(ScreenSize.width-75, ScreenSize.height-175);
//mainLabel.setSize(mainLabel.getSize());
mainLabel.setIcon(new ImageIcon(xychart.createBufferedImage(mainLabel.getWidth(),
mainLabel.getHeight())));
mainLabel.validate();

*/

```



```

setExtendedState(javax.swing.JFrame.MAXIMIZED_BOTH);

}

private void formWindowClosing(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:

}

private void abrirButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    JFileChooser readfilechooser = new JFileChooser();
    String readfile;
    int returnVal = readfilechooser.showOpenDialog(null);
    if (returnVal == JFileChooser.APPROVE_OPTION)
    {
        readfile=readfilechooser.getSelectedFile().getPath();
        //System.out.println("readfile:"+readfile);
        //System.out.println("fichero:"+fc.getSelectedFile().getName());
        //System.out.println("path:"+savefichero);

        try
        {
            File f;
            f=new File(readfile);
            if (!f.exists()) f.createNewFile();
            bufferedReader = new BufferedReader(new FileReader(f));
        }
        catch (IOException o){}

        double mediarssi=0;
        int paquetes=0;
        int numeropaquetes=0;
        long paso=Integer.parseInt(pasoTextField.getText());
        long anterior,actual,inicio;
        anterior=-1;
        actual=-1;
        inicio=-1;

        try
        {
            int i=0;

            serierrssi=new XYSeries("series rssi");
            seriepaquetes=new XYSeries("series paquetes");

            while((readline=bufferedReader.readLine()) != null)
            {
                splittedreadline=readline.split(";");

```



```

if (anterior == -1)
{
anterior=Long.parseLong(splittedreadline[0]);
inicio=anterior;
mediarssi=mediarssi+Double.parseDouble(splittedreadline[2]);
numeropaquetes++;
}
else
{
actual=Long.parseLong(splittedreadline[0]);
mediarssi=mediarssi+Double.parseDouble(splittedreadline[2]);
numeropaquetes++;
if ((actual - inicio) > paso)
{
serierssi.add(i,(mediarssi/numeropaquetes));
seriepaquetes.add(i,numeropaquetes);
mediarssi=0;
numeropaquetes=0;
inicio=actual;
i++;
}
anterior=actual;
}

}
bufferedReader.close();
}
catch (IOException e){System.out.println("tron"+e.getMessage());}

try{bufferedReader.close();} catch (Exception e){}

/*
xydataset = new XYSeriesCollection();
xydataset.addSeries(serierssi);
xydataset.addSeries(seriepaquetes);
*/

xydatasetpaquetes = new XYSeriesCollection();
xydatasetrssi = new XYSeriesCollection();
xydatasetpaquetes.addSeries(seriepaquetes);
xydatasetrssi.addSeries(serierssi);

xychart = ChartFactory.createXYLineChart(
    "Numero de paquetes y RSSI media (paso:"+paso+")", // chart title
    "Intervalos", // domain axis label
    "Numero de paquetes", // range axis label
    xydatasetpaquetes, // data

```



```

        PlotOrientation.VERTICAL,
        true, // include legend
        true,
        false);

    XYPlot plot = xychart.getXYPlot();
    NumberAxis axis2 = new NumberAxis("RSSI medio");
    axis2.setAutoRangeIncludesZero(false);
    plot.setRangeAxis(1, axis2);
    plot.setDataset(1, xydatasetrssi);
    plot.mapDatasetToRangeAxis(1, 1);

    XYItemRenderer renderer;
    renderer = plot.getRenderer();
    renderer.setSeriesPaint(0, Color.BLUE);

    StandardXYItemRenderer renderer2 = new StandardXYItemRenderer();
    renderer2.setSeriesPaint(0, Color.RED);
    plot.setRenderer(1, renderer2);

    /*
    final StandardXYItemRenderer renderer2 = new StandardXYItemRenderer();
    renderer2.setSeriesPaint(0, Color.black);
    plot.setRenderer(1, renderer2);
    */

    mainLabel.setSize(ScreenSize.width-75, ScreenSize.height-200);
    //mainLabel.setSize(mainLabel.getSize());
    mainLabel.setIcon(new ImageIcon(xychart.createBufferedImage(mainLabel.getWidth(),
mainLabel.getHeight())));
    mainLabel.validate();

    }
    else
    {
    System.out.println("error:");
    }
    }

    private void guardarButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        JFileChooser savefilechooser = new JFileChooser();
        String savefile;
        int returnVal = savefilechooser.showSaveDialog(null);
        if (returnVal == JFileChooser.APPROVE_OPTION)
        {
            savefile=savefilechooser.getSelectedFile().getPath()+".csv";
            //System.out.println("savefile:"+savefile);

```



```

try
{
File f;
f=new File(savefile);
if (!f.exists()) f.createNewFile();
bufferedWriter = new BufferedWriter(new FileWriter(f));

for (int i=0;i<serierssi.getItemCount();i++)
{
bufferedWriter.write(serierssi.getY(i)+" "+seriepaquetes.getY(i)+"");
bufferedWriter.newLine();
}

try {bufferedWriter.close();} catch (Exception e){}

}
catch (IOException o){}

}
}

private void poweroffActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
System.exit(0);
}

private void saveimageButtonActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
JFileChooser saveimagefilechooser = new JFileChooser();
String saveimagefile;
int returnVal = saveimagefilechooser.showSaveDialog(null);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
saveimagefile=saveimagefilechooser.getSelectedFile().getPath()+".jpg";
//System.out.println("savefile:"+savefile);
try {ChartUtilities.saveChartAsJPEG(new File(saveimagefile), xychart,
mainLabel.getWidth(), mainLabel.getHeight());} catch (Exception e){}
}
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new Graficas().setVisible(true);
}
});
}
}

```

## C.- Sniffer Atmel:

En el apartado 7.1.2 de este trabajo se comenta que para obtener datos sobre la calidad del enlace se intentó adquirir un sniffer inalámbrico que proporcionara datos sobre los paquetes enviados y recibidos por el canal.

Debido al precio de dichos dispositivos, la única opción viable encontrada fue el kit con el modelo RZRAVEN y el software para servicios inalámbricos AVR de Atmel.

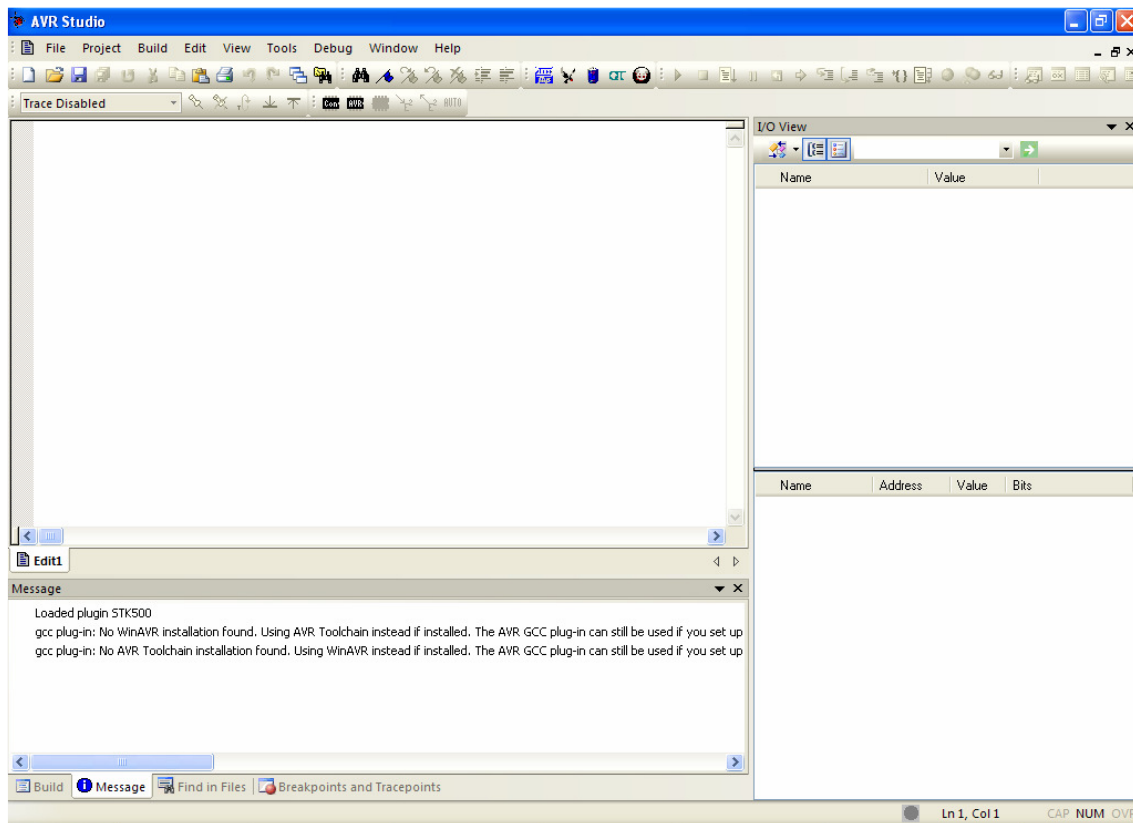


**Figura C.1.** Kit RZRAVEN de Atmel.

Mediante el software AVR Studio 4 se controlaba el dispositivo, que se conectaba directamente al PC donde se instalaba el software mediante un conector USB. En la figura C.2 se muestra una imagen del dispositivo, y en la figura C.3 una captura de pantalla del programa AVR Studio 4.



**Figura C.2.** Dispositivo RZRAVEN.



**Figura C.3.** Software AVR Studio 4.

Después de instalar el software y seguir todos los pasos indicados por el fabricante para poner en funcionamiento el dispositivo, aparecía en pantalla un mensaje de error.

Se comprobaron los requerimientos del sistema y se comprobó que todo estaba correcto. Se volvió a instalar el software y el hardware en el mismo PC, pero el problema persistía. Por tanto, se probó en otros 3 PCs diferentes, con distintas prestaciones y distintos sistemas operativos, pero no se logró solucionar el problema.

Después de contactar con el fabricante, y no pudiendo ellos solucionar el problema, se envió el kit adquirido para cambiarlo por otro, pensando que el recibido estaba defectuoso. Pero ni con un kit nuevo se consiguió solucionar el problema, por lo que se decidió desechar la opción de usar un sniffer inalámbrico para obtener datos de la calidad del enlace. En vez de esto, se crearon los programas Java descritos en el apartado 7.1 para obtener los datos acerca de los paquetes recibidos directamente de los sensores inalámbricos.





## **D.- Datos obtenidos en los escenarios:**

En varios de los escenarios analizados en este trabajo, los cuales se describen en el apartado 7, los datos obtenidos en cada una de las medidas se almacenaban en un documento de texto o en un documento excel. Si los datos se obtenían directamente de algún instrumento, como un osciloscopio, se almacenaban en un documento excel. Si por el contrario, los datos eran los recogidos por los programas Java, habitualmente se almacenaban en documentos de texto.

Debido al gran volumen de datos obtenidos (por ejemplo, existen muchas medidas de hasta 1.000.000 de datos, cada uno de ellos con tiempo de llegada, número de paquete y RSSI), incluir todos ellos en este documento tal cuál se obtuvieron es inviable. Por ello, en la carpeta llamada ANEXO D se adjuntan en su formato original.



# *ANÁLISIS TOPOLÓGICO DE REDES DE SENSORES INALÁMBRICOS EN ENTORNOS INTERIORES COMPLEJOS*



**Peio Lopez Iturri**

*Francisco Falcone Lanas*

*Carlos Fernández-Valdivielso*



# Índice



- 1- Introducción**
- 2- Objetivo**
- 3- Estado del arte**
- 4- Propagación**
- 5- Redes de sensores inalámbricos ZigBee**
- 6- Descripción de los dispositivos**
- 7- Resultados**
- 8- Conclusiones**
- 9- Líneas futuras**



## 1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras

# *1-Introducción*





## 1-Introducción

### 2-Objetivo

### 3-Estado del arte

### 4-Propagación

### 5-Redes de sensores inalámbricos ZigBee

### 6-Descripción de los dispositivos

### 7-Resultados

### 8-Conclusiones

### 9-Líneas futuras

- En la actualidad, el auge de las redes inalámbricas es una realidad.
- Las redes de sensores inalámbricos están presentes en un gran número de campos de aplicación, como son la monitorización industrial, monitorización de habitats, agricultura y ganadería, domótica, asistencia sanitaria, localización y seguimiento o seguridad y defensa, entre otros muchos.
- Hoy en día ZigBee es la tecnología más utilizada en este tipo de redes.
- Todas las redes y dispositivos analizados en este trabajo están basados en la norma IEEE 802.15.4 → ZigBee. Se caracterizan por estar los dispositivos alimentados por baterías.

1-Introducción

**2-Objetivo**

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras



# *2-Objetivo*





1-Introducción

**2-Objetivo**

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras

**El objetivo de este proyecto fin de carrera es  
analizar el impacto de entornos interiores  
complejos en el despliegue de redes de  
sensores inalámbricos.**

- Potencia recibida.
- Consumo de corriente (baterías).
- Calidad del enlace: PER (Packet Error Rate).
- Mediante:
  - o Simulación de trazado de rayos 3D.
  - o Medidas empíricas.



1-Introducción

2-Objetivo

**3-Estado del arte**

4-Propagación

5-Redes de sensores inalámbricos ZigBee

6-Descripción de los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras

# *3-Estado del arte*







1-Introducción

2-Objetivo

**3-Estado del arte**

4-Propagación

5-Redes de sensores inalámbricos ZigBee

6-Descripción de los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras

Empresas que ofrecen soluciones (módulo+software) basadas en los estándares 802.15.4 y ZigBee:

Empresa	Producto	Coste unidad (\$)
Digi	XBee-Pro	29,90
Atmel	ZigBit B0/A2	37,70
LS-Research	ProFLEX01	31,78
Cirronet	ZMN 2405/ZMN 2430	21,50 a 37
Panasonic	PAN4561	1: 58,06 → 250: 24,84



Chip:  
 Si el número de unidades es elevado, < 1 euro/unidad.



1-Introducción

2-Objetivo

**3-Estado del arte**

4-Propagación

5-Redes de sensores inalámbricos ZigBee

6-Descripción de los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras

- Empresas Españolas que implantan soluciones de redes de sensores inalámbricos:



- El COIT en 2010 organizó un ecosistema inicial de empresas especializadas en redes de sensores.

1-Introducción

2-Objetivo

3-Estado del arte

**4-Propagación**

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras



# *4-Propagación radioeléctrica*





1-Introducción

2-Objetivo

3-Estado del arte

**4-Propagación**

5-Redes de sensores inalámbricos ZigBee

6-Descripción de los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras

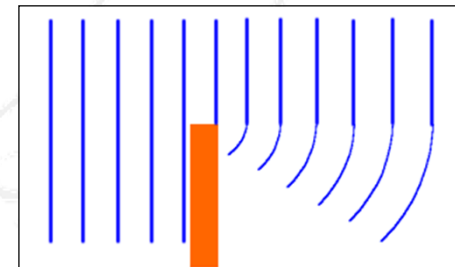
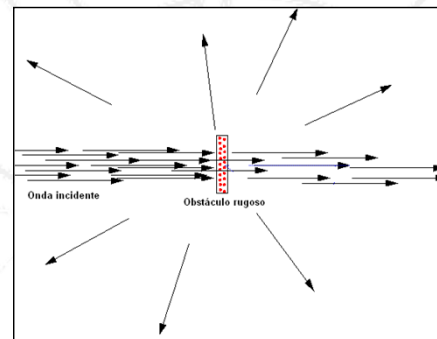
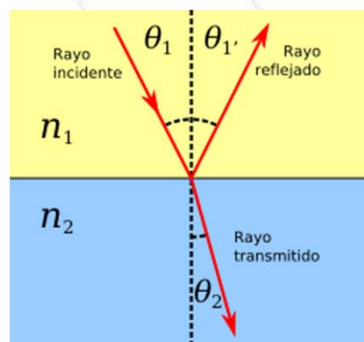
## 1.- Pérdidas de propagación

$$\frac{Pr}{Pt} = \frac{1}{4 \cdot \pi \cdot r^2} \cdot D_T \cdot A_{efR} = \left( \frac{\lambda}{4 \cdot \pi \cdot r} \right)^2 \cdot D_T \cdot D_R = \left( \frac{1}{\lambda \cdot r} \right)^2 \cdot A_{efT} \cdot A_{efR}$$

## 2.- Absorción (Refracción)

OBJETO	ATENUACIÓN
Pared de escayola	3 dB
Pared de cristal con armazón metálico	6 dB
Pared de ladrillos	4 dB
Ventana de oficina	3 dB
Puerta metálica	6 dB
Puerta metálica en pared de ladrillos	12 dB
Cuerpo humano	3 dB

## 3.- Reflexión, Dispersión y Difracción





1-Introducción

2-Objetivo

3-Estado del arte

**4-Propagación**

5-Redes de sensores inalámbricos ZigBee

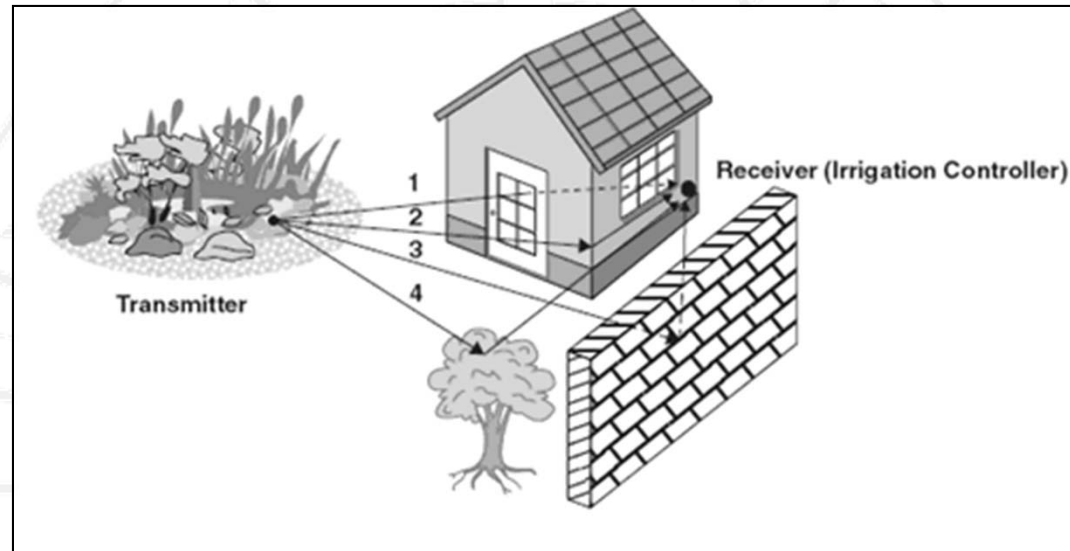
6-Descripción de los dispositivos

7-Resultados

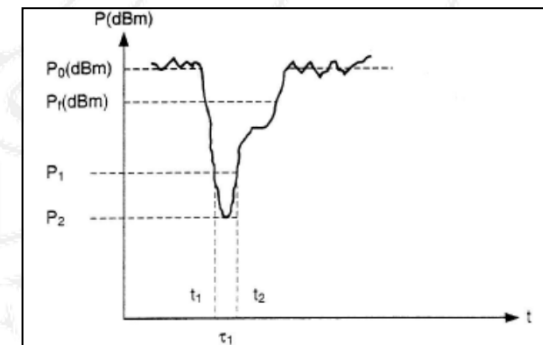
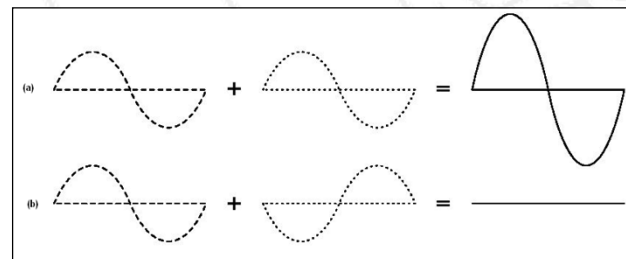
8-Conclusiones

9-Líneas futuras

## Propagación multirayecto



## - Desvanecimientos multirayecto:





1-Introducción

2-Objetivo

3-Estado del arte

**4-Propagación**

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

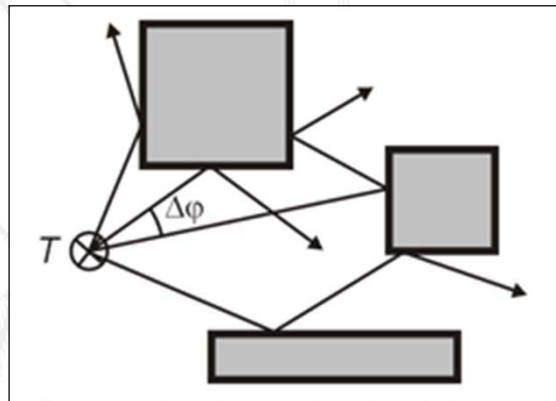
7-Resultados

8-Conclusiones

9-Líneas futuras

## Modelos de propagación en interiores: Trazado de rayos

- Software desarrollado enteramente en la UPNA.
- Basado en Óptica Geométrica → Aprox: Energía en forma de rayo.
- Modelo determinista → Ecuaciones de Maxwell.



1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

**5-Redes de  
sensores  
inalámbricos  
ZigBee**

6-Descripción de  
los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras



# *5-Redes de sensores inalámbricos ZigBee*

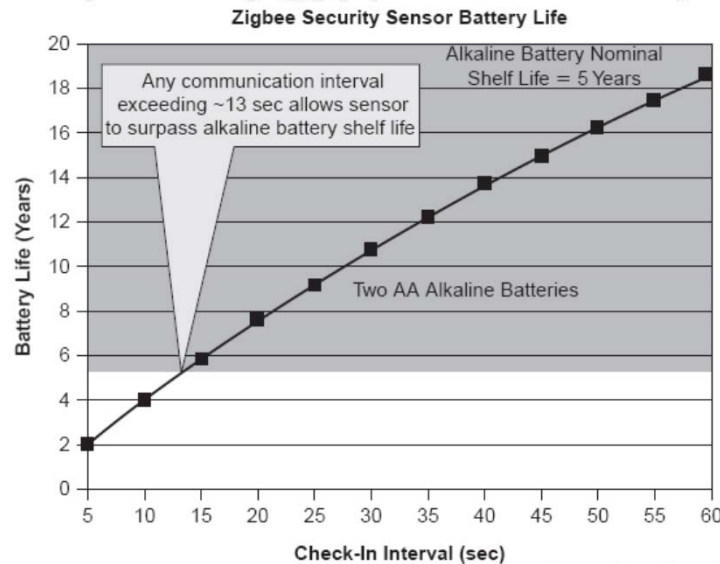
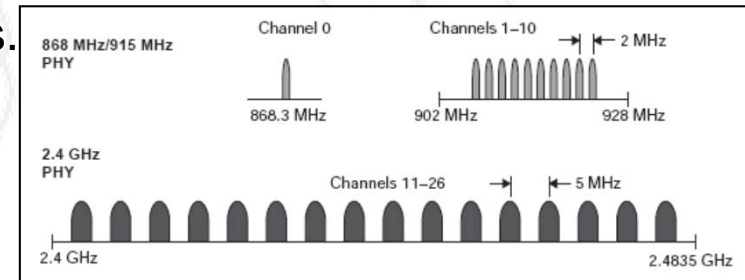


- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados
- 8-Conclusiones
- 9-Líneas futuras

**ZigBee** es el nombre del estándar más utilizado para la implementación de redes inalámbricas de:



- Bajo coste.
- Baja tasa de transmisión de datos.
- Corto alcance.
- Bajo consumo de energía.



	Frequency (MHz)	Number of Channels	Modulation	Chip Rate (Kchip/s)	Bit Rate (Kb/s)	Symbol Rate (Ksymbol/s)	Spreading Method
	868-868.6	1	BPSK	300	20	20	Binary DSSS
	902-928	10	BPSK	600	40	40	Binary DSSS
Optional	868-868.6	1	ASK	400	250	12.5	20-bit PSSS
	902-928	10	ASK	1600	250	50	5-bit PSSS
Optional	868-868.6	1	O-QPSK	400	100	25	16-array orthogonal
	902-928	10	O-QPSK	1000	250	62.5	16-array orthogonal
	2400-2483.5	16	O-QPSK	2000	250	62.5	16-array orthogonal





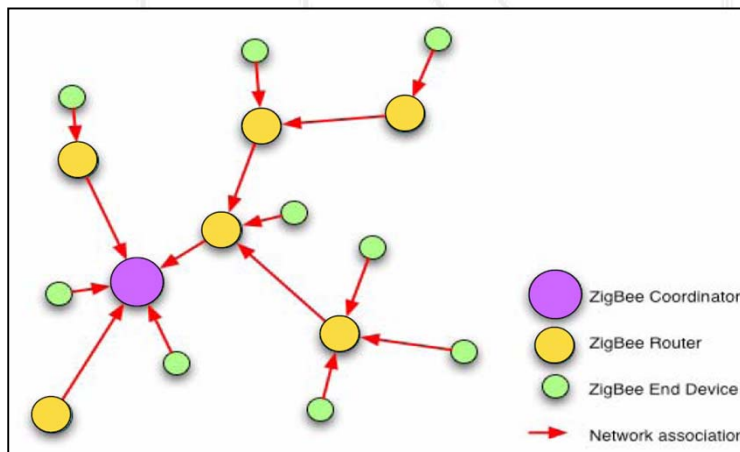
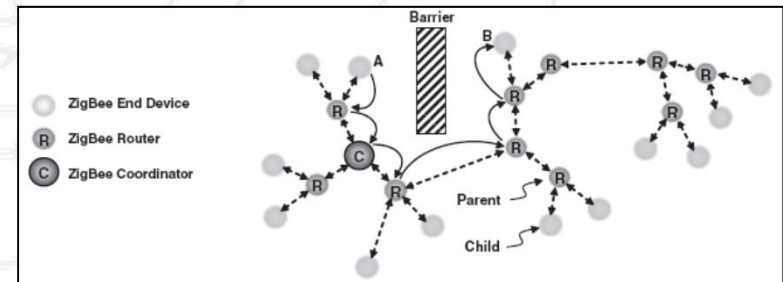
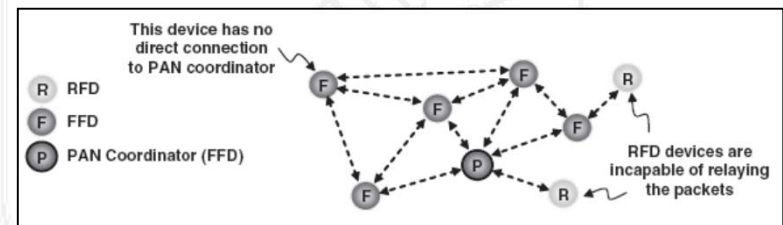
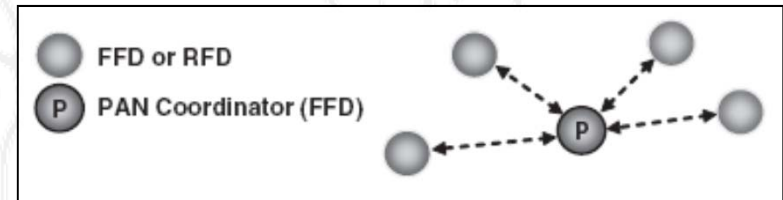
- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee**
- 6-Descripción de los dispositivos
- 7-Resultados
- 8-Conclusiones
- 9-Líneas futuras

## Tipos de nodos ZigBee:

- **Coordinador ZigBee (ZC):**
  - Forma la red.
  - Decide incorporaciones a la red.
  - Enrutamiento.
- **Router ZigBee (ZR):**
  - Enrutamiento.
  - Extiende cobertura.
- **Dispositivo final ZigBee (ZED):**
  - Modo ahorro de energía.
  - El más simple de los tres.

## Topologías de red:

- Estrella.
- Malla.
- Árbol.





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

**6-Descripción  
de los  
dispositivos**

7-Resultados

8-Conclusiones

9-Líneas futuras

# *6-Descripción de los dispositivos*





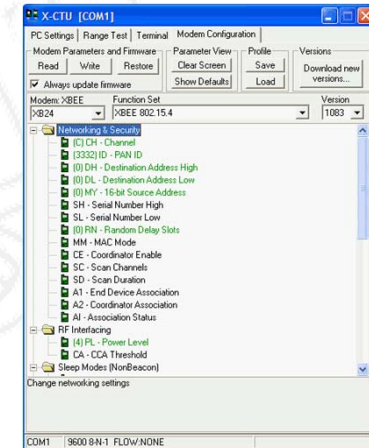
- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos**
- 7-Resultados
- 8-Conclusiones
- 9-Líneas futuras

## XBee-Pro de Digi International

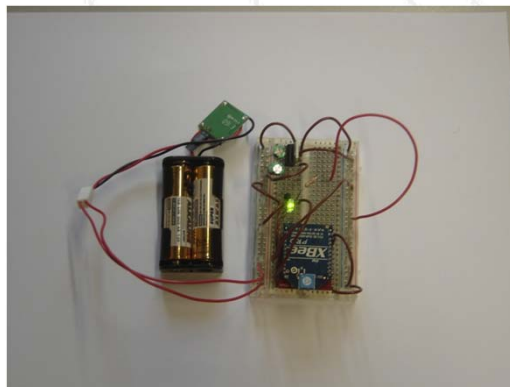


### Características:

- Banda ISM 2.4 GHz.
- Alcance en entornos interiores o urbanos: 300 metros.
- Pot. máx. = 18 dBm (63 mW).
- Sensibilidad: -100 dBm.
- Vel. transmisión máx. = 250 Kbps.
- Posibilidad ACKs.
- Topologías en malla, árbol y estrella.
- Bajo consumo de energía:
  - Consumo corriente máx. = 250 mA.



Software de configuración X-CTU.



Parameter Range: 0 - 7 (standard rates)  
 0x80-0x3D090 (non-standard rates up to 250 Kbps)

Parameter	Configuration (bps)
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600
7	115200

Default Parameter Value: 3



1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

**6-Descripción  
de los  
dispositivos**

7-Resultados

8-Conclusiones

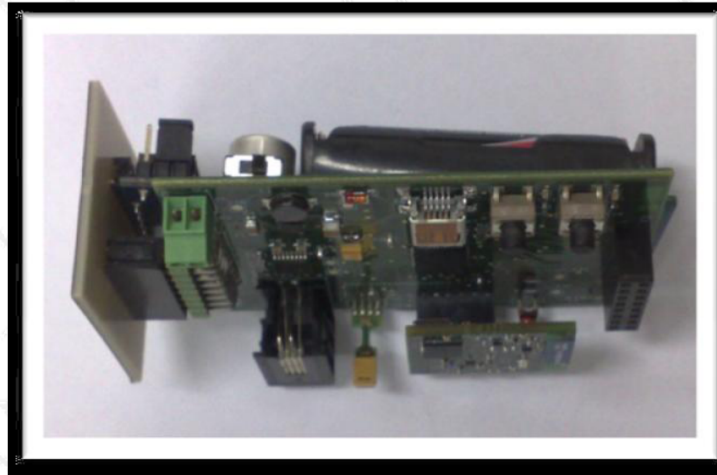
9-Líneas futuras

## ■ CyFi Universidad de Valencia

Características transceiver: Cypress CyFi CYRF7936



- Banda ISM 2.4 GHz.
- Pot. máx. = 4 dBm.
- Sensibilidad máx.: -97 dBm.
- Vel. transmisión máx. = 250 Kbps.
- Bajo consumo de energía:
  - Consumo de corriente (transmitiendo -5 dBm) = 21 mA.
  - Consumo de corriente (sleep mode) = 15 mA.





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

**7-Resultados**

8-Conclusiones

9-Líneas futuras

# *7-Resultados*

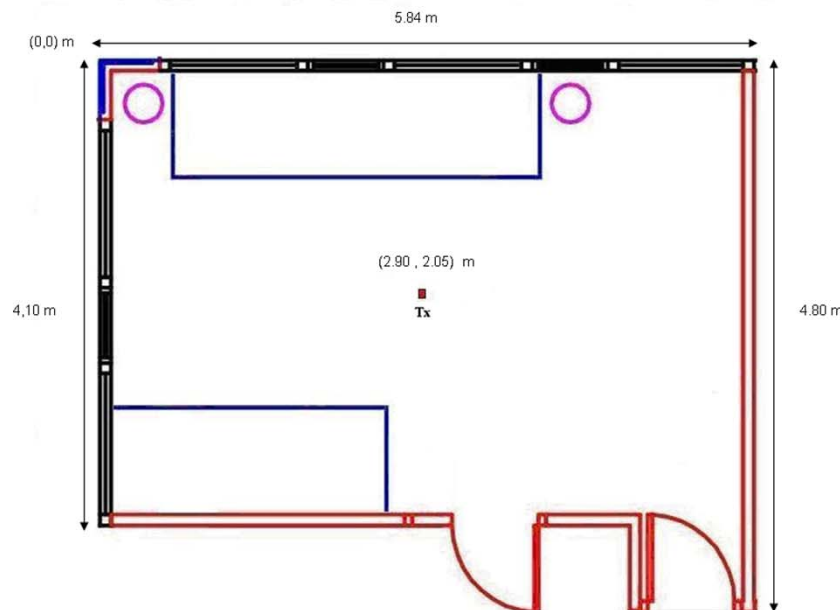




- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados**
- 8-Conclusiones
- 9-Líneas futuras

## Escenario 1: RSSI (Received Signal Strength Indicator)

- Distribución de potencia dentro de un escenario interior.
- Edificio Jerónimo de Ayanz, UPNA. Laboratorio L5 y despacho D5.
- XBee-Pro, a 18 dBm (máx.).
- Analizador de espectros portátil Agilent FieldFox.





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de sensores inalámbricos ZigBee

6-Descripción de los dispositivos

**7-Resultados**

8-Conclusiones

9-Líneas futuras

## Escenario 1: RSSI (Received Signal Strength Indicator)

(x,y)	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5
0.5	-40.33	-36.37	-32.23	-36.35	-35.34	-36.27	-40.05	-48.74	-39.52	-39.99	-42.98
1	-35.73	-31.76	-32.56	-29.36	-31.62	-31.65	-38.59	-42.93	-44.15	-35.99	-45.96
1.5	-43.17	-37.51	-27.42	-26.16	-26.53	-29.71	-43.64	-44.59	-39.11	-49.49	-42.31
2	-36.18	-33.53	-36.03	-29.77	-23.07	Tx	-41.81	-44.76	-46.96	-46.09	-45.13
2.5	-44.60	-51.07	-43.36	-39.02	-45.41	-22.37	-25.75	-31.40	-37.85	-39.92	-40.79
3	-44.48	-37.61	-40.27	-42.92	-40.10	-26.71	-29.58	-30.51	-32.54	-49.87	-42.08
3.5	x	-53.36	-49.66	-42.77	-31.69	-32.18	-29.86	-34.82	-32.89	-34.18	-43.35
4	x	x	x	x	x	x	x	x	x	-50.29	-36.95
4.5	x	x	x	x	x	x	x	x	x	-37.40	-41.55

### Conclusiones:

- A mayor distancia, menor potencia recibida.
- No siempre se cumple: Muchos obstáculos → Propagación multirrayecto.



1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

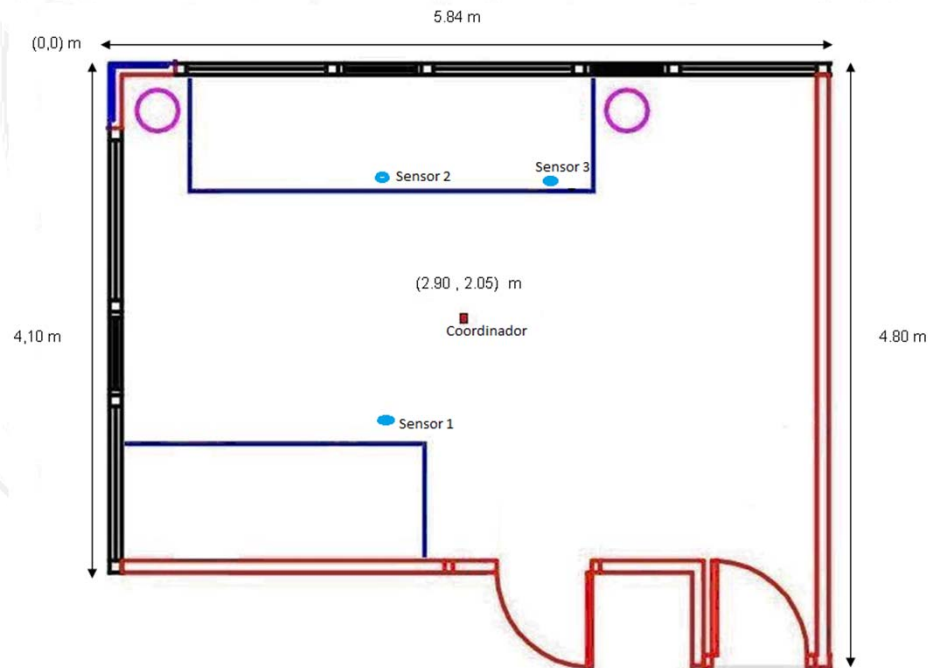
**7-Resultados**

8-Conclusiones

9-Líneas futuras

## Escenario 2: RSSI + Consumo con varios transmisores

- Comportamiento de una red con un coordinador y con varios dispositivos transmitiendo simultáneamente.
- Edificio Jerónimo de Ayanz, UPNA. Laboratorio L5.
- XBee-Pro.
- RSSI: Mediante software.
- Consumo: Mediante osciloscopio DPO3014 de Tektronix.







1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

**7-Resultados**

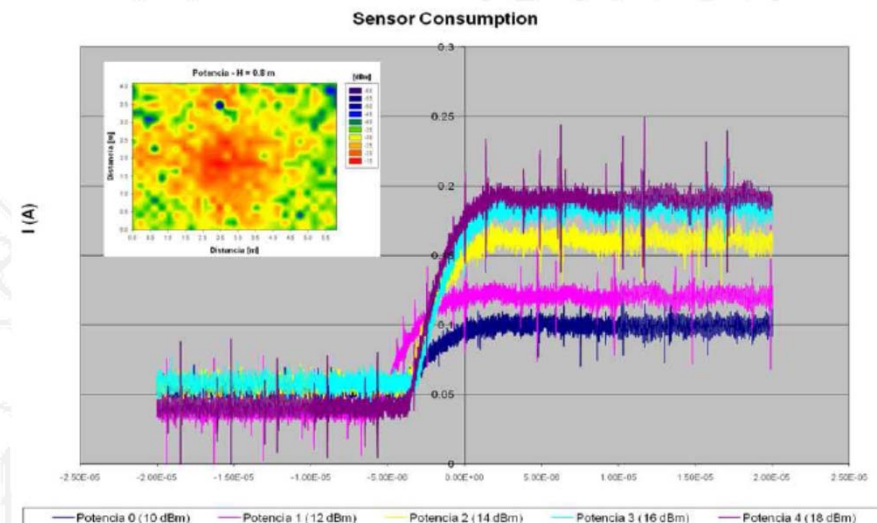
8-Conclusiones

9-Líneas futuras

## Escenario 2: RSSI + Consumo con varios transmisores

### Conclusiones:

- Propagación multitrayecto + personas → variaciones RSSI.
- Consumo de corriente depende del nivel de potencia transmitida:
  - 18 dBm (máx.) → 200 mA.
  - 10 dBm (mín.) → 45 mA.





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

**7-Resultados**

8-Conclusiones

9-Líneas futuras

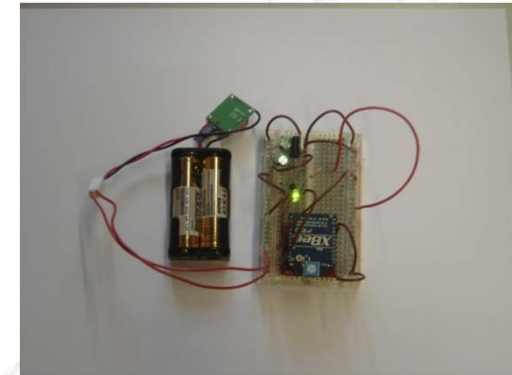
## Escenario 3: RSSI con diferentes obstáculos

- Variaciones RSSI con diferentes tipos de obstáculos.
- Edificio Jerónimo de Ayanz, UPNA. Laboratorio L5.
- XBee-Pro, a 18 dBm.
- RSSI: Mediante software.
- 1 transmisor → 1 receptor.



### Atenuaciones medidas:

- Armario de plástico: 3 dB.
- Cajón de madera: 5 dB.
- Caja de cartón: 0 dB.
- Cubierta de plástico: 3-4 dB.
- Funda de tela: 10 dB.



### Conclusiones:

- Las atenuaciones de los obstáculos (absorción) son datos importantes a tener en cuenta.
- Orientación antena + cables = hasta 15 dBm.



1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

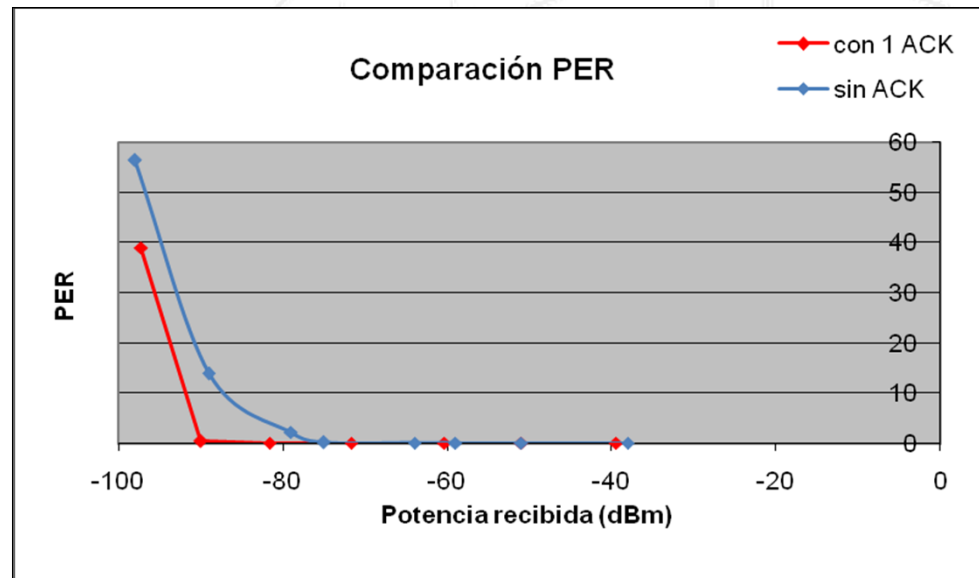
6-Descripción de  
los dispositivos

**7-Resultados**

8-Conclusiones

9-Líneas futuras

## Escenario 4: PER (sin ACK vs. ACK)



- Edificio Jerónimo de Ayanz.
- XBee-Pro, a 18 dBm.
- 1 transmisor → 1 receptor.
- 500.000 paquetes TX

### Conclusiones:

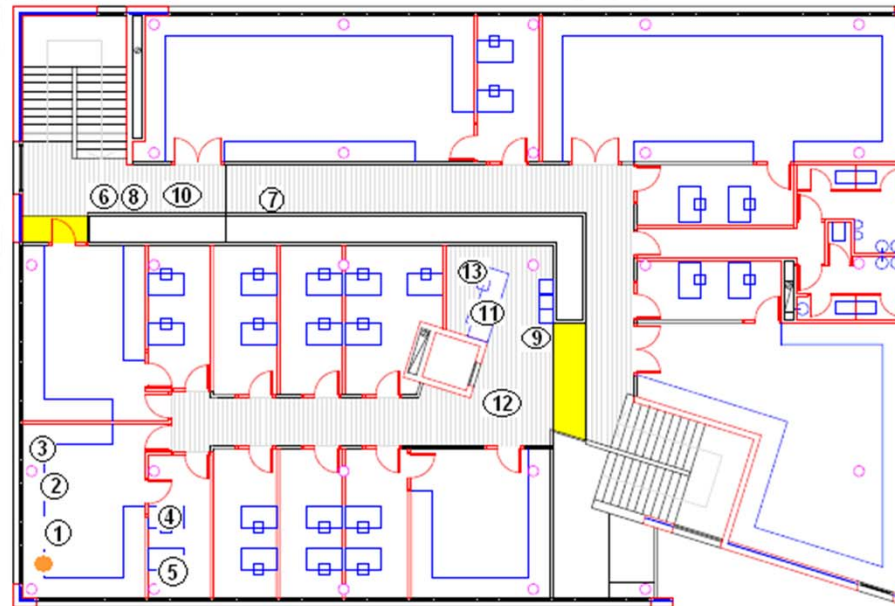
- La curva comienza a crecer exponencialmente a -90 dBm (antes -70 dBm).
- Mejora importante con 1 ACK.



- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados**
- 8-Conclusiones
- 9-Líneas futuras

## Escenario 5: PER (1.000.000)

- Edificio Jerónimo de Ayanz, UPNA.
- XBee-Pro, a 18 dBm.
- 1 transmisor → 1 receptor.
- 1.000.000 paquetes transmitidos, sin ACK.



### Altura (metros):

- 1: 0.95
- 2: 0.95
- 3: 0.95
- 4: 0.74
- 5: 0.74
- 6: 0.25
- 7: 0.25
- 8: 0.25
- 9: 0.25
- 10: 0.25
- 11: 0.25
- 12: 0.25
- 13: 0.25
- 14: 0.25
- 15: 0.25
- 16: 0.25
- 17: 0.25
- 18: 0.25

● Transmisor



1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

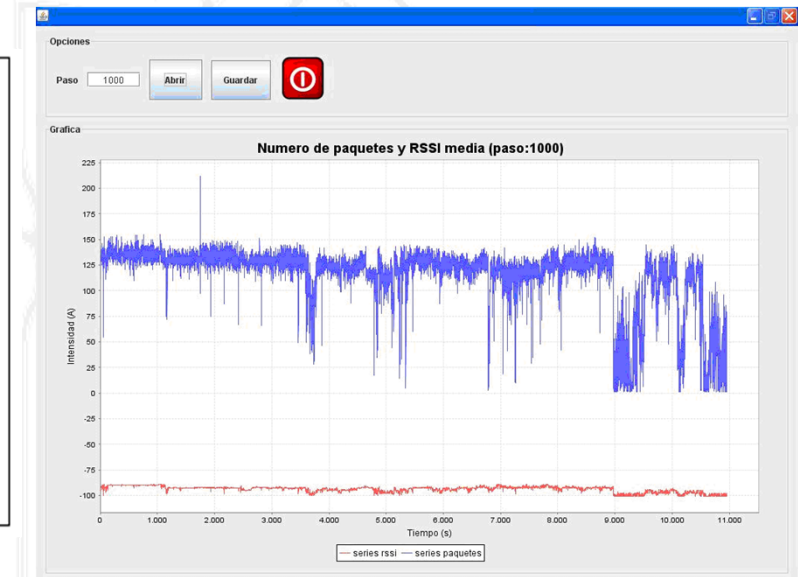
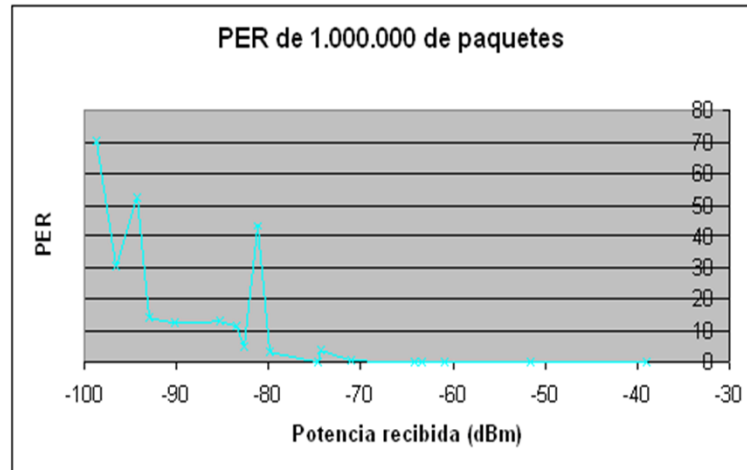
6-Descripción de  
los dispositivos

**7-Resultados**

8-Conclusiones

9-Líneas futuras

## Escenario 5: PER (1.000.000)



### Conclusiones:

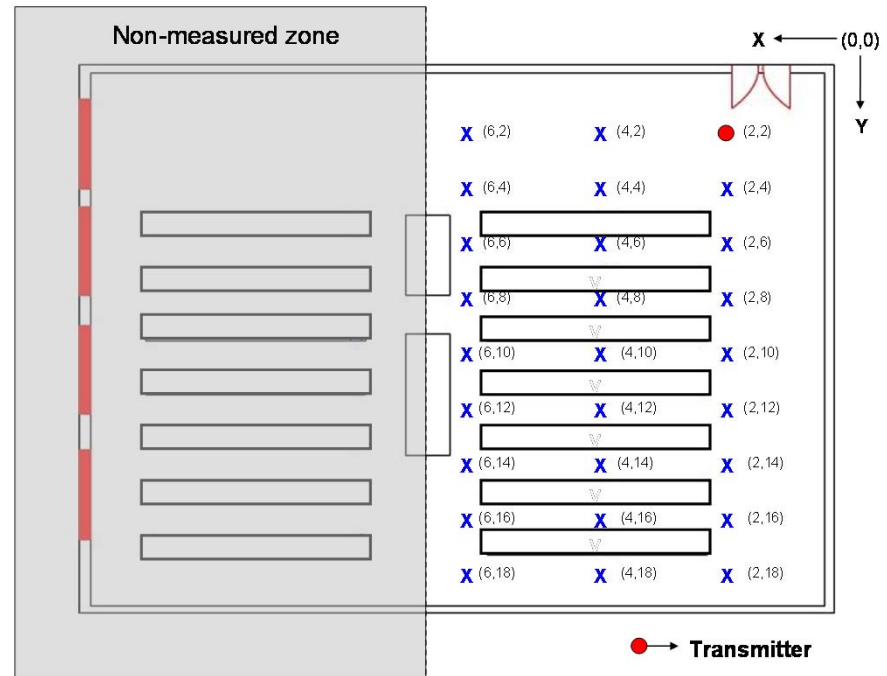
- Mayor cantidad de paquetes → Datos más robustos.
- El tránsito humano afecta fuertemente a la señal, sobre todo para valores de potencia recibida cercanos al nivel de sensibilidad.



- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados**
- 8-Conclusiones
- 9-Líneas futuras

## Escenarios 6: Dispositivos CyFi

- Laboratorio de Radiocomunicaciones, UPNA.
- CyFi, a 4 dBm.
- Estudio: RSSI, consumo de corriente y PER.





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
 sensores  
 inalámbricos  
 ZigBee

6-Descripción de  
 los dispositivos

**7-Resultados**

8-Conclusiones

9-Líneas futuras

## Escenarios 6: Dispositivos CyFi

	Coordenada X		
Coordenada Y	6	4	2
2	-19	-19	Tx
4	-15	-18	-18
6	-20	-19	-21
8	-31	-31	-31
10	-31	-31	-31
12	-23	-31	-31
14	-12	-23	-21
16	-24	-21	-17
18	-24	-17	-25

### Resultados RSSI:

- 1 paquete cada 20 segundos.
- Capacidad de autorregulación

	Coordenada X		
Coordenada Y	6	4	2
2	-15.5	-14.5	Tx
4	-13	-12.5	-14.71
6	-12*	-13	-19.82
8	-14.36	-12*	-14.4
10	0	0	-11*
12	0	0	0
14	0	0	0
16	0	0	0
18	0	0	0

### Resultados RSSI:

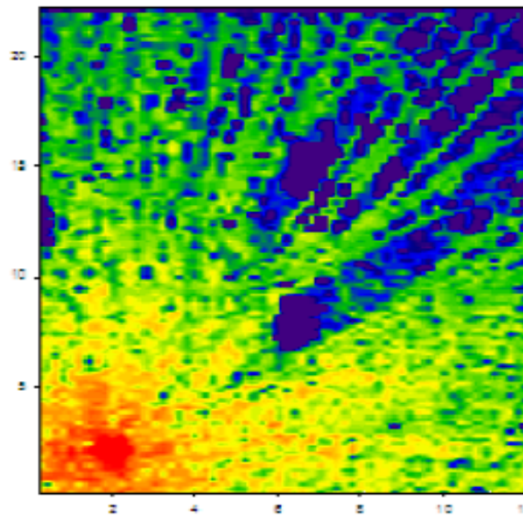
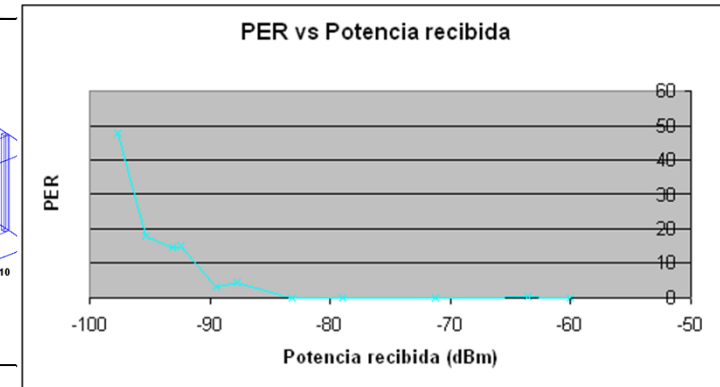
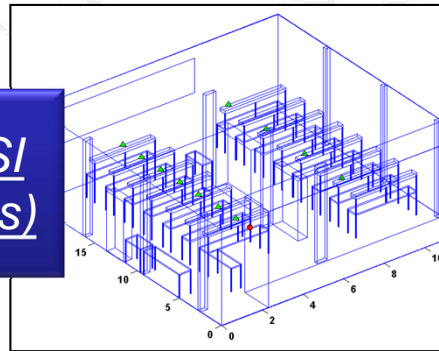
- 50 paquetes/s. (\* 5 paquetes/s).
- Mayor velocidad → poco alcance.



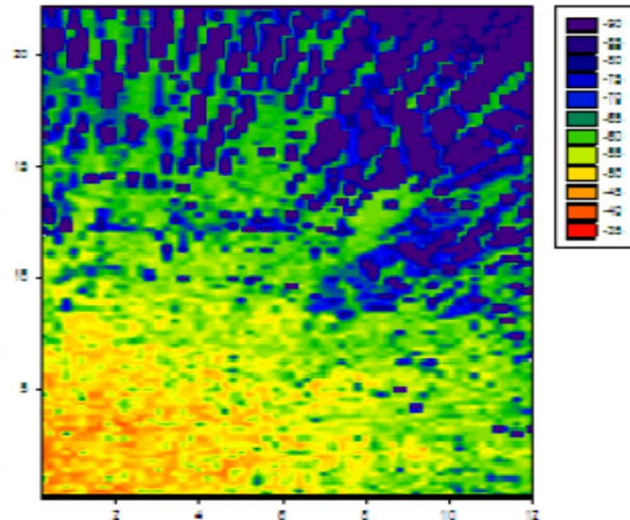
- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados**
- 8-Conclusiones
- 9-Líneas futuras

## Escenarios 6: Dispositivos CyFi

Resultados RSSI  
(trazado de rayos)



**Altura = 0,81 m**



**Altura = 2,3 m**

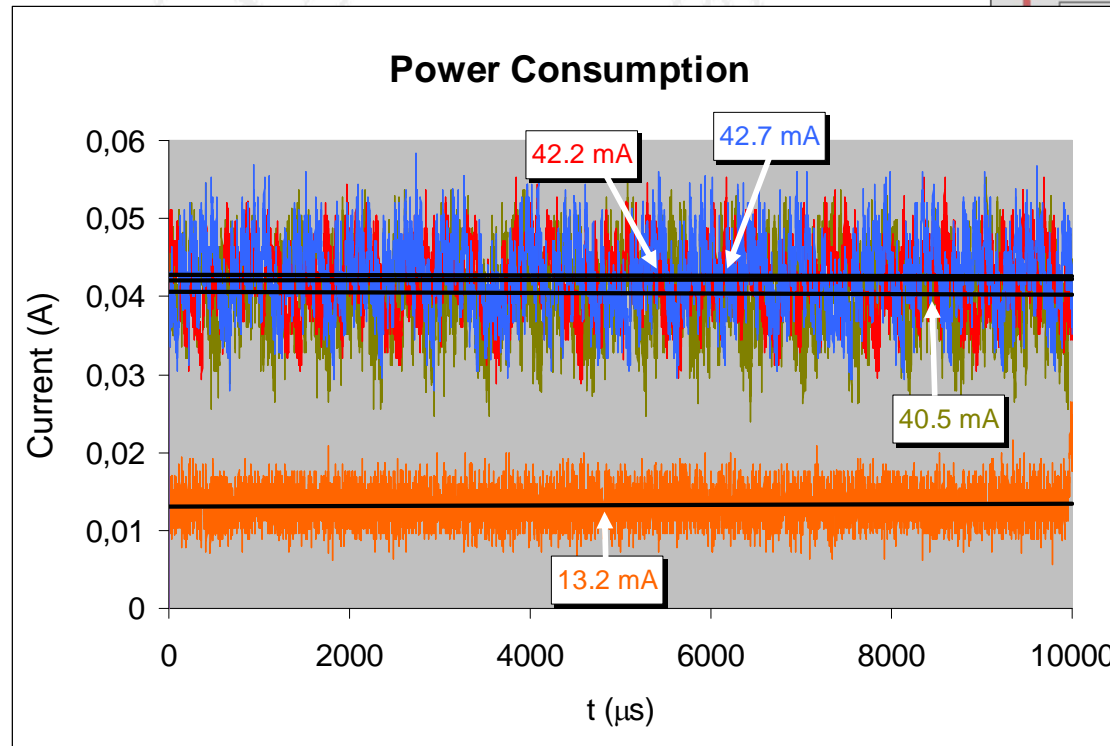
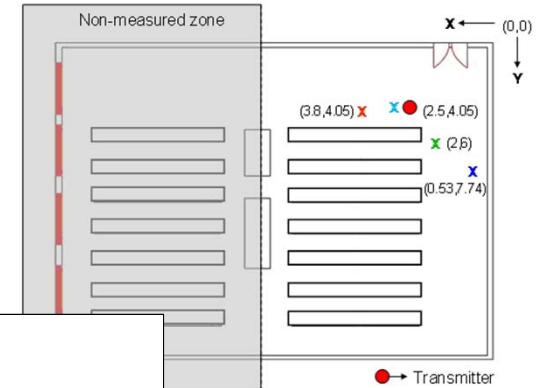




- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados**
- 8-Conclusiones
- 9-Líneas futuras

## Escenarios 6: Dispositivos CyFi

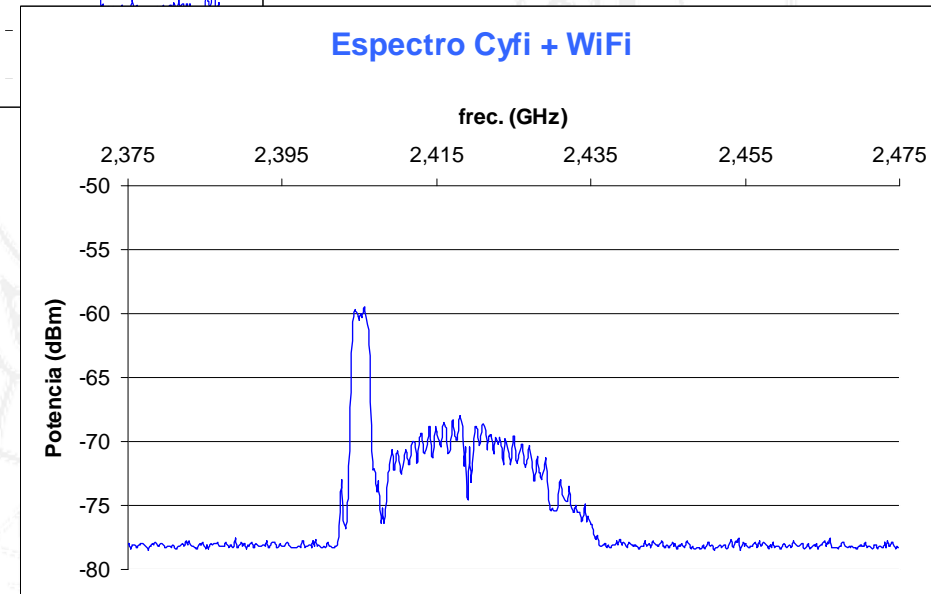
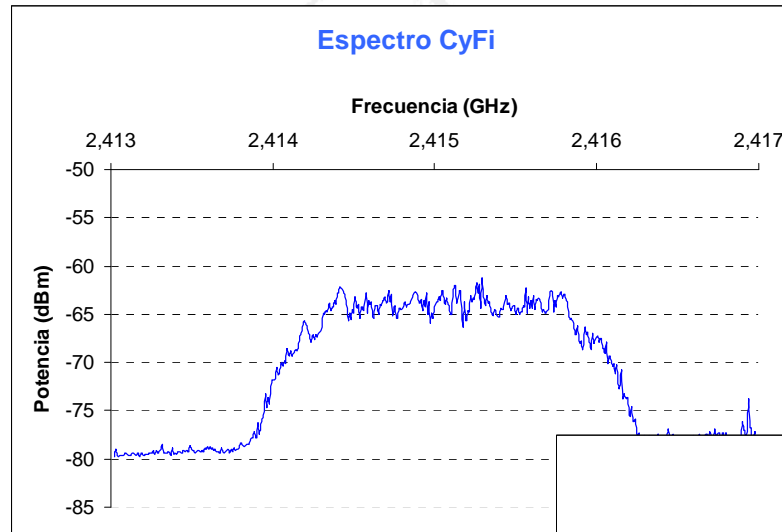
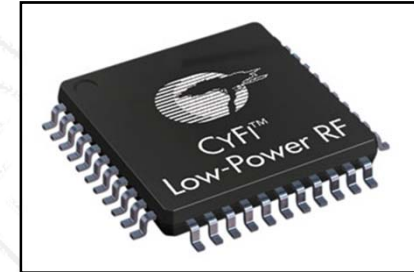
### Consumos:





- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados**
- 8-Conclusiones
- 9-Líneas futuras

## Escenarios 7: Interferencia WiFi





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de sensores inalámbricos ZigBee

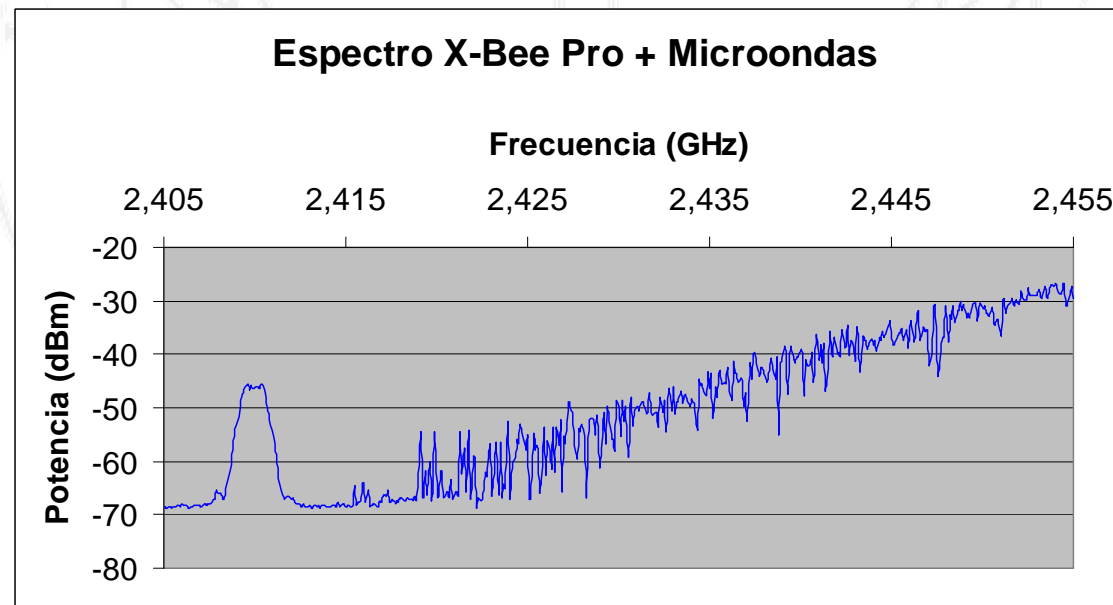
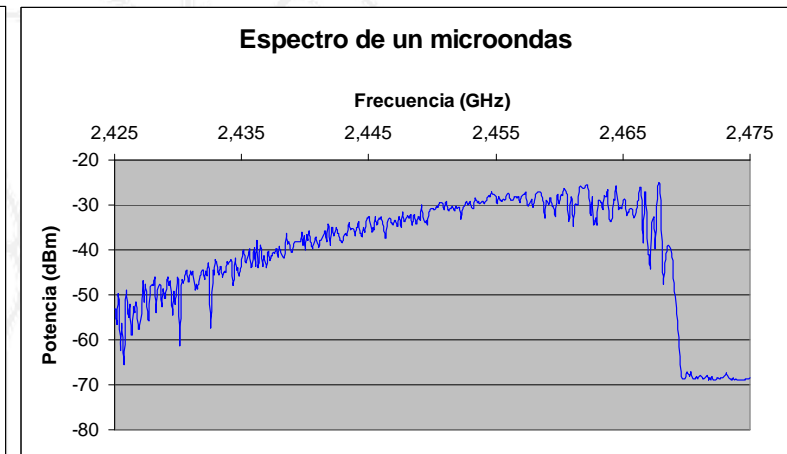
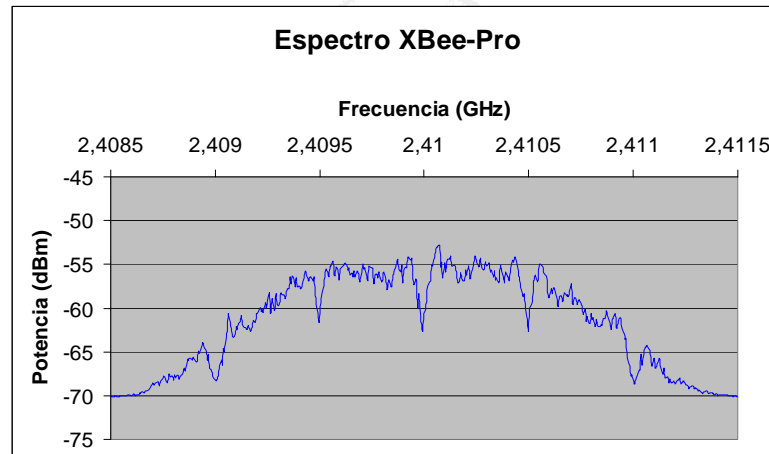
6-Descripción de los dispositivos

**7-Resultados**

8-Conclusiones

9-Líneas futuras

## Escenarios 8: Interferencia de un microondas



- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados**
- 8-Conclusiones
- 9-Líneas futuras

## Escenarios 9: Movilidad

- Primer escenario dinámico.

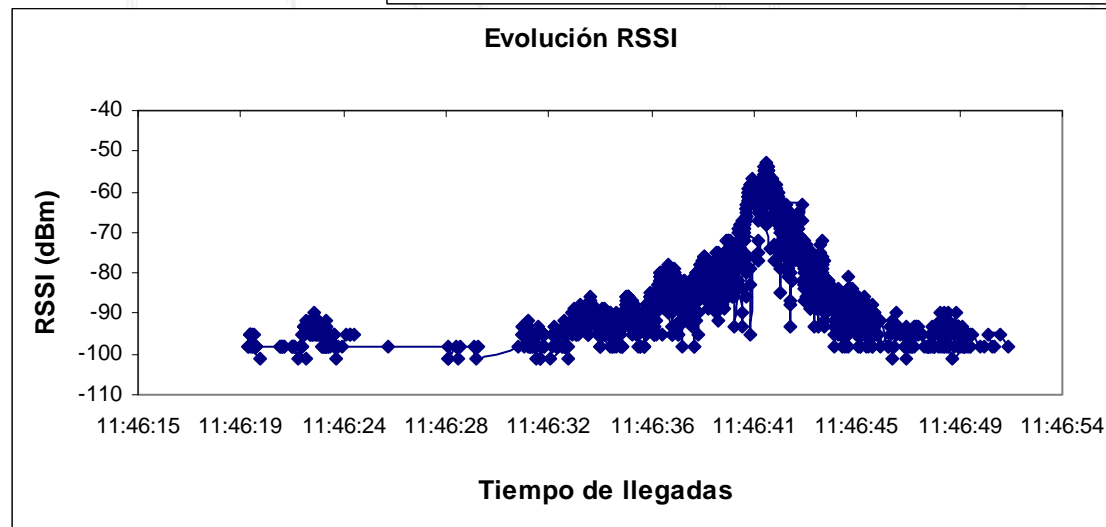
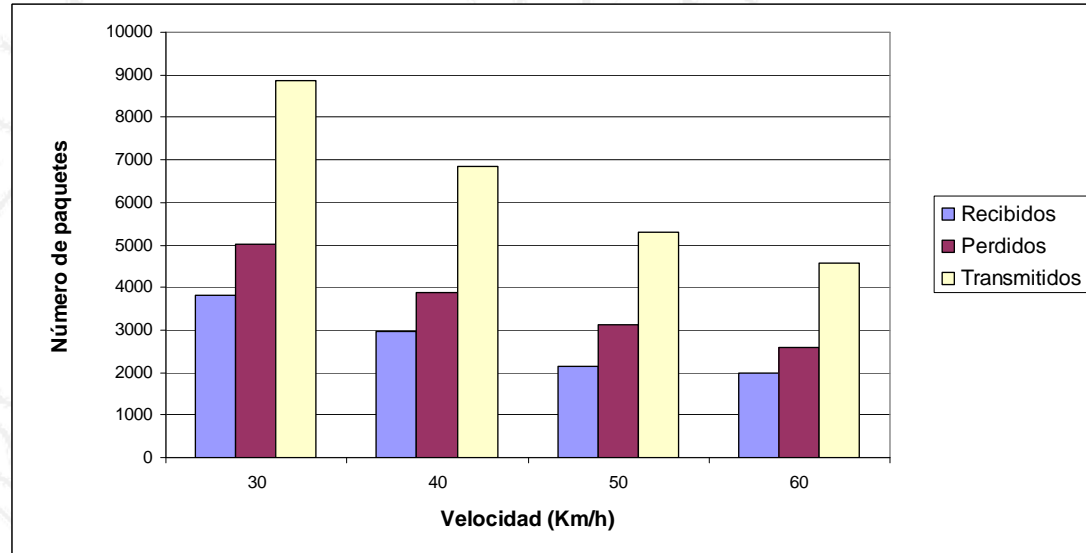




- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados**
- 8-Conclusiones
- 9-Líneas futuras

## Escenarios 9: Movilidad

### - Resultados:





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

7-Resultados

**8-Conclusiones**

9-Líneas futuras

# *8-Conclusiones*





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

7-Resultados

**8-Conclusiones**

9-Líneas futuras

## **Conclusiones:**

- La radiopropagación en escenarios interiores es compleja, donde el nivel de potencia de señal recibida depende fuertemente de la topología, afectando igualmente a otros parámetros como el PER y el consumo de corriente.**
- El PER está directamente relacionado con la potencia recibida por los dispositivos inalámbricos.**
- ZigBee es muy robusto con respecto a la pérdida de paquetes.**
- Realizando una buena planificación de la topología y la morfología de red, así como la elección de los dispositivos inalámbricos más adecuados, el consumo de potencia y el rendimiento global del sistema pueden ser optimizados considerablemente.**



1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

7-Resultados

**8-Conclusiones**

9-Líneas futuras

## **Conclusiones:**

- El software de trazado de rayos proporciona estimaciones de potencia recibida bastante buenas, permitiendo realizar análisis más completos.
- Los escenarios dinámicos son mucho más complejos que los estáticos, y requieren de un estudio mayor.
- Existen otras tecnologías (WiFi) y electrodomésticos como los microondas, que pueden interferir en el correcto funcionamiento de una red de sensores inalámbricos.





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

7-Resultados

8-Conclusiones

**9-Líneas  
futuras**

# *9-Líneas futuras*





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

5-Redes de  
sensores  
inalámbricos  
ZigBee

6-Descripción de  
los dispositivos

7-Resultados

8-Conclusiones

**9-Líneas  
futuras**

## **Líneas futuras:**

- Mayor cantidad de medidas y simulaciones (ACKs, niveles potencia de transmisión,...).
- Aumentar el número de sensores.
- Distintos dispositivos inalámbricos.
- Interferencias WiFi y microondas.
- Escenarios más complejos (movilidad, dispositivos en cuerpo humano, entorno doméstico, hospitales, hoteles,...)



- 1-Introducción
- 2-Objetivo
- 3-Estado del arte
- 4-Propagación
- 5-Redes de sensores inalámbricos ZigBee
- 6-Descripción de los dispositivos
- 7-Resultados
- 8-Conclusiones
- 9-Líneas futuras

# *Congresos y publicaciones*





1-Introducción

2-Objetivo

3-Estado del arte

4-Propagación

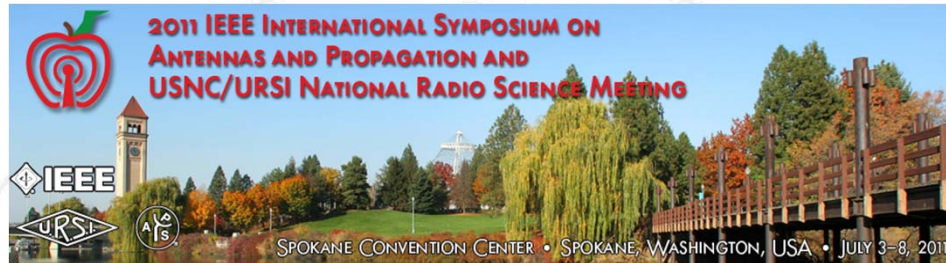
5-Redes de sensores inalámbricos ZigBee

6-Descripción de los dispositivos

7-Resultados

8-Conclusiones

9-Líneas futuras



Título del trabajo:

***Analysis of Topological impact in Wireless Indoor Sensor Networks.***

Autores:

***Peio Lopez, Juan Antonio Nazabal, Víctor Torres, Carlos Fernández-Valdivielso, Francisco Falcone.***

**URSI**  
**leganés 2011**  
XXVI Edición de URSI



UNIÓN CIENTÍFICA  
INTERNACIONAL DE RADIO

Título del trabajo:

***Análisis del comportamiento de redes de sensores inalámbricas en entornos interiores heterogéneos.***

Autores: ***Peio Lopez et al.***



***ANÁLISIS TOPOLÓGICO DE REDES DE  
SENSORES INALÁMBRICOS  
EN ENTORNOS INTERIORES COMPLEJOS***

***¡Muchas  
gracias!***

**Peio Lopez Iturri**

**Pamplona, 8 de Septiembre de 2011**

**Francisco Falcone Lanas**

**Carlos Fernández-Valdivielso**

