

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN



FINAL THESIS REPORT

Ingeniero en Telecomunicación

DEVELOPMENT OF A PLATFORM WITH  
MULTIPLE RF INTERFACES FOR COGNITIVE  
WIRELESS SENSOR NETWORKS

Agustín Tena García

Septiembre 2013

Documento maquetado con TEXIS v.1.0+.

Este documento está preparado para ser impreso a doble cara.

# FINAL THESIS REPORT

**Título:** IMPLEMENTACIÓN, ANÁLISIS Y EVALUACIÓN DE UN ALGORITMO DE LOCALIZACIÓN PARA REDES INALÁMBRICAS DE SENSORES

**Autor:** AGUSTÍN TENA GARCÍA

**Tutor:** ELENA ROMERO PERALES

**Departamento:** DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

## MIEMBROS DEL TRIBUNAL

**Presidente:** D. ÁLVARO DE GUZMAN FERNÁNDEZ GONZÁLEZ

**Vocal:** D. ALVARO ARAUJO PINTO

**Secretario:** D. PEDRO JOSÉ MALAGÓN MARZO

**Suplente:** D. MIGUEL ÁNGEL SÁNCHEZ GARCÍA

## CALIFICACIÓN:

Se certifica que Ramiro Utrilla Gutiérrez ha completado con éxito su Trabajo Fin de Máster obteniendo así la titulación de Máster Oficial en Ingeniería de Sistemas Electrónicos de la ETSI Telecomunicación, Universidad Politécnica de Madrid, España.

Madrid, a \_\_\_\_\_ de \_\_\_\_\_ de



# DEVELOPMENT OF A PLATFORM WITH MULTIPLE RF INTERFACES FOR COGNITIVE WIRELESS SENSOR NETWORKS

AUTOR: Agustín Tena García

TUTOR: Elena Romero Perales



Departamento de Ingeniería Electrónica  
Escuela Técnica Superior de Ingenieros de Telecomunicación  
Universidad Politécnica de Madrid

Septiembre 2013



*A mis padres y  
a mi hermano.*



# Agradecimientos

*Lo esencial nunca está en la partitura.*

George Steiner

A todas las personas que forman parte del Laboratorio de Aplicaciones Bioacústicas por su apoyo, información, orientación, asesoramiento y acompañamiento a lo largo de este proyecto, confiando plenamente en mí y en mi toma de decisiones. A Michel por darme esta oportunidad y a Joan por su tutorización, ayuda, y por todo el tiempo y consejos que me ha dedicado. Gracias también a: María, Marta, Mike, Serge, Alex y Ludwig, que habéis hecho del laboratorio un segundo hogar. Sin duda me habéis impresionado; no solo como grupo de investigación, también, y sobre todo, por vuestro valor humano. Recordaré este año en el LAB con mucho cariño.

A José M. Miguel por todo el apoyo que me ha prestado durante este año en la UPC, primero como profesor y luego como director ponente. Por tus ánimos y consejos, y por esa cercanía con tus alumnos. Hace tiempo leí una frase que decía: *El cerebro aprende por emociones*. No dejes de emocionarnos a todos, como hasta ahora, con la radiofrecuencia.

A todos mis profesores y compañeros de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid (EPS-UAM). Ha sido un placer compartir estos años con vosotros. Mención especial para Javi y Pablo, por estar siempre codo con codo, tanto académica como personalmente, y a Luis, cuya amistad ha dejado una huella imborrable en mí.

A la gente de la Oficina de Relaciones Internacionales y Secretaría, tanto de la EPS-UAM como de la UPC, por toda su amabilidad y ayuda con los trámites.

A todos mis compañeros de piso, aquí en Barcelona. ¡Qué gran familia! ¡Cuántos buenos momentos juntos! Y los que vendrán... Ésto solo es un *j hasta pronto!*

A mis amigos en Madrid y a mi gente del Grupo Scout Estrella Polar junto a los que aprendí a querer hacer de este planeta un mundo mejor. Nombrarlos a todos sería imposible. He crecido junto a vosotros y, desde hace mucho, ya formáis parte de mí, de lo que soy... Allá donde esté, siempre vendréis conmigo.

A Elena por estar a mi lado y hacerme feliz cada día.

A mis padres y a mi hermano que me lo han dado todo. Gracias por vuestra eterna paciencia en mi educación, por haberme sabido guiar en mi vida sin dirigirme, y por todo vuestro cariño, amor y apoyo incondicional.

Lo esencial no está en la partitura, está en el músico, en todas las personas que me habéis acompañado a lo largo de todo este tiempo haciendo posible mi crecimiento

personal y académico. Sin duda, yo no podría ser sin vosotros. Gracias a todos.

# Resumen

La contaminación acústica generada por la actividad humana en el mar presenta efectos nocivos a nivel físico, fisiológico y de comportamiento sobre la fauna marina. El Laboratorio de Aplicaciones Bioacústicas (LAB) de la Universidad Politécnica de Cataluña (UPC), dentro del proyecto LIDO, trabaja en el desarrollo de herramientas de control y mitigación de estos efectos. Concretamente emplea técnicas de Monitorización mediante Acústica Pasiva (PAM) para la detección, clasificación y localización de fuentes acústicas. Con el fin de adquirir información del medio marino mediante hidrófonos, el LAB plantea la utilización de una red de boyas autónomas capaces de enviar sus datos a través de otras boyas intermedias hasta una estación colectora.

Este proyecto, comprendido dentro del proyecto LIDO, tiene como objetivo diseñar y realizar un primer prototipo de red multisalto especialmente pensada para su operación en el mar. El diseño del mismo va dirigido a soportar un servicio de envío de alarmas, orientado a la detección de la presencia de cetáceos en una determinada área de actuación; y a buscar la operatividad del sistema logrando una distancia entre boyas de algunos kilómetros, y un consumo energético reducido.

Para alcanzar este objetivo se han desarrollado las siguientes actividades: Realización de un estudio de posibilidades tecnológicas orientado a la elección de los equipos. Definición de la arquitectura del sistema para integrar correctamente los nuevos módulos con los ya desarrollados por el LAB. Elaboración de un estudio teórico de un enlace radioeléctrico en el medio marino comparándolo con medidas reales de alcance en el mismo medio realizadas con los equipos adquiridos. Implementación y simulación del software necesario para la gestión de entrega de alarmas al nodo colector.

Los principios rectores del proyecto han sido: La coherencia y el equilibrio para el desarrollo sostenible en el medio marino. El compromiso de aportar buenas prácticas mediante el estudio, la investigación e innovación en lo que a herramientas de control y mitigación del impacto medioambiental se refiere. El respeto a la normativa vigente en materia de utilización del espectro radioeléctrico.

La ejecución integrada de este Plan de Acción ha dado lugar a un prototipo de red ad hoc de datos que servirá como herramienta para continuar experimentando y trabajando en esta línea. La red opera en la banda de frecuencia de 868 MHz. En las pruebas de alcance sobre la superficie del mar para un enlace simple se han logrado buenas prestaciones hasta una distancia aproximada de 2 kilómetros. Se ha verificado la correcta entrega de alarmas para una red formada por: tres nodos, un colector y dos generadores, colocados de manera alineada en tierra firme, formando un enlace bidireccional en el que los extremos requerían del nodo central para su comunicación.

**PALABRAS CLAVE:** *red ad hoc de datos, monitorización en entorno marino, protocolo de encaminamiento AODV, banda de 868 MHz.*

# Abstract

Noise pollution generated by human activity in the sea produces harmful effects on the marine fauna in a physical, physiological and behavioral way. The Laboratory of Applied Bio-Acoustics (LAB) at the Technical University of Catalonia (UPC) joining the LIDO project, works on the development of tools to control and mitigate these effects. They employ Passive Acoustic Monitoring (PAM) techniques for the detection, classification and location of acoustic sources. In order to obtain information about marine environment using hydrophones, the LAB is planning to use an autonomous buoy network capable of sending data through intermediate buoys to a base station.

This project, included in the LIDO project, aims to design and implement a first prototype of a multihop network specially designed to operate in the sea. This design is orientated to hold up an alarm service focused on detecting the presence of cetaceans in a specific area; in order to achieve this goal, the system must properly operate within buoys distanced some kilometers with minimal energy consumption.

To achieve this objective, several tasks were performed: First, a technological study was carried out in order to select the appropriate equipment. Second, a definition of the system's architecture was established to allow us to insert the new modules with the ones already made by the LAB. To continue, an elaboration of a theoretical study of a radioelectric link in marine environment and a comparison of the results with empirical measures made with the acquired equipment in the same conditions were made. Finally, an implementation and simulation of the necessary software for the alarm delivery management to the collector node was developed.

The foundations of this project have been: Coherence and balance for a sustainable development of the marine environment. The commitment of providing best practices through the study, investigation and innovation of the environmental impact control and mitigation tools. All this taking in consideration the current radioelectric spectrum legislation.

The integrated implementation of this plan has led to a data ad hoc network prototype which will be used as a helpful tool for further investigation in this field. The network operates at 868 MHz. In the test range over the sea surface for a single link we have achieved a good performance in a range of 2 kilometers. It has been verified that the correct alarm delivery for a network is composed of three nodes, one collector and two generators that are located on the ground in line, forming a duplex link in which both ends needed the central node to communicate.

**KEY WORDS:** *ad hoc data network, marine environment monitoring, AODV routing protocol, 868 MHz frequency band.*



# Índice

<b>Agradecimientos</b>	<b>IX</b>
<b>Resumen</b>	<b>XI</b>
<b>Abstract</b>	<b>XIII</b>
<b>I Capítulos</b>	<b>1</b>
<b>1. Introducción</b>	<b>3</b>
1.1. Motivación . . . . .	3
1.2. Marco de trabajo . . . . .	6
1.2.1. Laboratorio de Aplicaciones Bioacústicas . . . . .	6
1.2.2. Proyecto LIDO . . . . .	7
1.3. Objetivos . . . . .	10
1.4. Metodología de trabajo . . . . .	11
1.5. Organización del documento . . . . .	13
<b>2. Fundamentos y estado del arte</b>	<b>15</b>
2.1. Bandas de frecuencia . . . . .	15
2.2. Redes ad hoc . . . . .	18
2.3. Redes de malla inalámbricas . . . . .	19
2.4. Redes de sensores inalámbricas . . . . .	20
2.5. Radioenlaces en entornos marinos . . . . .	21
2.5.1. Fundamentos básicos de propagación . . . . .	21
2.5.2. El entorno marino . . . . .	23
2.6. Wi-Fi . . . . .	24
2.7. WiMAX . . . . .	27
2.8. ZigBee . . . . .	29
2.9. Otras tecnologías . . . . .	31
<b>3. Definición del prototipo de red ad hoc</b>	<b>33</b>
3.1. Factores determinantes . . . . .	33
3.2. Características deseables . . . . .	35
3.3. Elección de una tecnología . . . . .	36
3.4. Visión global del sistema LIDO DCL implementado sobre boyas autónomas . . . . .	39

3.5. Arquitectura del sistema LIDO DCL sobre una boyá autónoma para operar dentro de una red ad hoc . . . . .	40
<b>4. Hardware</b>	<b>43</b>
4.1. Análisis de mercado y elección del equipo . . . . .	43
4.2. XBee-PRO®868 RF Modules . . . . .	46
4.2.1. Descripción . . . . .	46
4.2.2. Software X-CTU . . . . .	48
4.2.3. Primeras pruebas . . . . .	49
4.3. Estudio teórico de un enlace radioeléctrico en el mar . . . . .	51
4.3.1. Constitución del modelo . . . . .	51
4.3.2. Balance de un enlace radioeléctrico . . . . .	52
4.3.3. Enlace radioeléctrico en condiciones de espacio libre . . . . .	53
4.3.4. Influencia de la superficie sobre la que discurre la onda . . . . .	54
4.3.5. Modelo de propagación de tierra plana . . . . .	56
4.4. Implementación de un enlace simple de datos en el mar . . . . .	61
4.4.1. Escenario . . . . .	61
4.4.2. Procedimiento . . . . .	63
4.4.3. Medidas . . . . .	64
4.4.4. Análisis de probabilidad . . . . .	66
4.5. Discusión de resultados . . . . .	66
<b>5. Software</b>	<b>71</b>
5.1. Módulo de entrega remota de datos . . . . .	71
5.1.1. Envío, enrutado y acuse de recibo de alarmas . . . . .	72
5.1.2. Localización de nodos . . . . .	72
5.2. Protocolos de encaminamiento . . . . .	73
5.2.1. Protocolos de encaminamiento proactivos, reactivos e híbridos .	73
5.2.1.1. Protocolos de encaminamiento proactivo . . . . .	74
5.2.1.2. Protocolos de encaminamiento reactivo . . . . .	74
5.2.1.3. Protocolos de encaminamiento híbridos . . . . .	74
5.2.1.4. Elección de un protocolo de encaminamiento . . . . .	75
5.2.2. Protocolo AODV . . . . .	76
5.2.2.1. Visión general . . . . .	76
5.2.2.2. Declaración de aplicabilidad . . . . .	77
5.2.2.3. Tabla de rutas y tabla de RREQ's . . . . .	77
5.2.2.4. Formato de mensajes . . . . .	78
5.2.2.5. Descubrimiento de Ruta . . . . .	80
5.2.2.6. Mantenimiento de Ruta . . . . .	83
5.2.3. Protocolos basados en el AODV . . . . .	84
5.2.3.1. AODVjr . . . . .	84
5.2.3.2. AODVbis . . . . .	85
5.2.3.3. LoWPAN-AODV . . . . .	85
5.2.3.4. LOAD . . . . .	85
5.2.3.5. TinyAODV . . . . .	86
5.2.3.6. NST-AODV . . . . .	86

5.2.4. Definición del protocolo de encaminamiento AODV-LAB . . . . .	87
5.2.5. Comparación entre los distintos protocolos de encaminamiento .	88
5.3. Implementación en C/Linux . . . . .	88
5.3.1. Introducción y justificación . . . . .	88
5.3.2. Organización del código . . . . .	89
5.3.3. Tramas . . . . .	90
5.3.3.1. Trama RREQ . . . . .	92
5.3.3.2. Trama RREP . . . . .	93
5.3.3.3. Trama DATA . . . . .	93
5.3.3.4. Trama ACK . . . . .	94
5.3.3.5. Trama PREQ . . . . .	95
5.3.3.6. Trama PREP . . . . .	96
5.3.4. Estructura y flujo de ejecución del programa . . . . .	96
5.3.4.1. Primer chequeo: Alarms . . . . .	98
5.3.4.2. Segundo chequeo: Puerto serie . . . . .	98
5.3.4.3. Tercer chequeo: Cola de mensajes . . . . .	98
5.3.4.4. Cuarto chequeo: Envío de PREQ . . . . .	100
5.4. Validación y resultados . . . . .	100
5.4.1. Método de simulación . . . . .	101
5.4.2. Escenario 1 . . . . .	104
5.4.3. Escenario 2 . . . . .	110
5.4.4. Escenario 3 . . . . .	117
5.4.5. Escenario 4 . . . . .	122
5.4.6. Conclusiones . . . . .	125
<b>6. Prueba final del prototipo de red ad hoc de datos</b>	<b>129</b>
6.1. Introducción . . . . .	129
6.2. Primera prueba . . . . .	130
6.3. Segunda prueba . . . . .	133
6.4. Tercera prueba . . . . .	137
6.5. Discusión de resultados . . . . .	139
<b>7. Conclusiones y trabajo futuro</b>	<b>141</b>
7.1. Conclusiones . . . . .	141
7.2. Trabajo futuro . . . . .	142
<b>II Apéndices</b>	<b>145</b>
<b>A. Presupuesto</b>	<b>147</b>
<b>B. Código MATLAB</b>	<b>149</b>
B.1. freeSpace.m . . . . .	149
B.2. twoRayModel.m . . . . .	150
B.3. measurements.m . . . . .	152
<b>C. Cabecera y definición de funciones</b>	<b>155</b>

C.1. extractMessInfo . . . . .	155
C.2. checkAlarm . . . . .	155
C.3. sendLocalAlarm . . . . .	156
C.4. exeDATA . . . . .	156
C.5. sendDATA . . . . .	156
C.6. exeACK . . . . .	157
C.7. sendACK . . . . .	157
C.8. serialData2queue . . . . .	157
C.9. initSeqNum . . . . .	157
C.10. updateSeqNum . . . . .	158
C.11. checkRREQ . . . . .	158
C.12. exeRREQ . . . . .	158
C.13. sendRREQ . . . . .	159
C.14. checkPREQ . . . . .	159
C.15. exePREQ . . . . .	159
C.16. sendPREQ . . . . .	160
C.17. exeRREP . . . . .	160
C.18. sendRREP . . . . .	161
C.19. exePREP . . . . .	161
C.20. sendPREP . . . . .	161
C.21. checkNewRoute . . . . .	162
C.22. newRoute . . . . .	162
C.23. getRoute . . . . .	162

# Índice de figuras

1.1. Fuentes de contaminación acústica marina producidas por actividades humanas. . . . .	4
1.2. Niveles de sonido y frecuencias de fuentes sonoras antropogénicas y naturales del medio marino. . . . .	4
1.3. Ejemplos del impacto ambiental generado por el ser humano en el medio marino. . . . .	6
1.4. Detalles de la interfaz pública del proyecto LIDO. . . . .	8
1.5. Ejemplos de topologías de red. . . . .	9
1.6. Diagrama de la metodología de trabajo. . . . .	11
1.7. Comunicación con los compañeros del LAB. . . . .	12
2.1. Regiones definidas por la UIT en su Reglamento de Radiocomunicaciones.	17
2.2. Posibles implementaciones de una red de sensores inalámbrica (WSN). .	20
2.3. Zona de Fresnel. . . . .	22
2.4. Logo Wi-Fi. . . . .	25
2.5. Logo WiMAX Forum. . . . .	27
2.6. Topología Multihop Relay de WiMAX. . . . .	27
2.7. Imágenes tomadas del proyecto TRITON. (NICT©) . . . . .	28
2.8. Pila de protocolos ZigBee. . . . .	30
2.9. Resultados obtenidos en Zennaro et al., 2010. . . . .	31
3.1. Tipos de boyas según el tamaño. . . . .	34
3.2. Arquitectura de la implementación del sistema LIDO DCL sobre una boya autónoma.	40
3.3. Arquitectura del sistema LIDO DCL sobre una boya autónoma para operar dentro de una red ad hoc. . . . .	41
4.1. Vistas del módulo XBee-PRO 868. . . . .	47
4.2. Conversor UART-USB y nodo final formado por el XBee-PRO 868, la antenna y el conversor UART-USB. . . . .	47
4.3. Pestañas del software X-CTU de Digi International. . . . .	49
4.4. Módulo configurable en <i>loopback</i> mediante un <i>jumper</i> . . . . .	50
4.5. Valores de mínimo y máximo de RSSI medidos con la aplicación <i>Range Test</i> . . . . .	51
4.6. Modelo general simplificado de un sistema de radiocomunicación. . . .	52
4.7. Pérdida básica de propagación en condiciones de espacio libre en función de la distancia, para una frecuencia de operación $f = 869,525\text{MHz}$ . .	53

4.8. Potencia disponible a la entrada del receptor en función de la distancia en condiciones de espacio libre, para una frecuencia de operación $f = 869,525 MHz$ . . . . .	54
4.9. Esquema de rayos del modelo de propagación de tierra plana. . . . .	57
4.10. Pérdida básica de propagación descrita por el modelo de tierra plana. .	59
4.11. Pérdida básica de propagación de varios modelos. . . . .	60
4.12. Potencia disponible a la entrada del receptor en función de la distancia utilizando el modelo de tierra plana. . . . .	61
4.13. Enlaces implementados en las medidas de campo. . . . .	62
4.14. Nodos en los distintos enlaces. . . . .	65
4.15. Medidas realizadas de potencia disponible en el receptor para en enlace 1.	67
4.16. Medidas hipotéticas de potencia recibida en los enlaces 2 y 3. . . . .	68
5.1. Formato del mensaje RREQ del protocolo AODV. . . . .	79
5.2. Formato del mensaje RREP del protocolo AODV. . . . .	80
5.3. Formato del mensaje RERR del protocolo AODV. . . . .	80
5.4. Proceso de Descubrimiento de Ruta en una red ad hoc que utiliza el protocolo de encaminamiento AODV para enviar datos desde un nodo origen S hasta un nodo destino D. . . . .	82
5.5. Arquitectura de operación del módulo de entrega remota de datos. . . .	90
5.6. Organización del código implementado. . . . .	91
5.7. Estructura y flujo de ejecución del programa. . . . .	97
5.8. Diagrama de flujo de la parte de chequeo de alarmas. . . . .	99
5.9. Diagrama de flujo de la parte de chequeo del puerto serie. . . . .	99
5.10. Diagrama de flujo de la parte de chequeo de la cola de mensajes. . . . .	100
5.11. Diagrama de flujo de la parte de envío de mensajes PREQ. . . . .	101
5.12. Simulación de un enlace bidireccional entre dos nodos. . . . .	102
5.13. Simulación de enlaces bidireccionales entre tres nodos. . . . .	103
5.14. Escenario 1 de simulación: Enlace bidireccional entre dos nodos. . . .	104
5.15. Escenario 2 de simulación: Enlace bidireccional entre tres nodos alineados.	111
5.16. Formación del camino de vuelta del nodo D al S. . . . .	112
5.17. Formación del camino de ida del nodos S al D. . . . .	113
5.18. Actualización de tablas de encaminamiento. . . . .	114
5.19. Escenario 3 de simulación: Varias rutas distintas entre dos nodos. . . .	117
5.20. Descubrimiento de la ruta más corta entre los nodos S y D. . . . .	118
5.21. Evolución de las tablas de encaminamiento del nodo S. . . . .	118
5.22. Evolución de las tablas de encaminamiento del nodo S. . . . .	119
5.23. Tabla de encaminamiento del nodo D. . . . .	120
5.24. Tabla de encaminamiento del nodo D. . . . .	121
5.25. Escenario 4 de simulación: Red dinámica. . . . .	122
5.26. Tabla de encaminamiento del nodo C. . . . .	123
5.27. Tabla de encaminamiento del nodo D. . . . .	124
5.28. Tabla de encaminamiento del nodo D. . . . .	124
6.1. Escenario de pruebas: Enlace bidireccional entre tres nodos alineados. .	130

# Índice de Tablas

2.1.	Bandas ICM definidas por la UIT . . . . .	16
2.2.	Notas de Utilización Nacional de las Bandas ICM definidas en el CNAF 2010 . . . . .	18
2.3.	Tipos de Wi-Fi . . . . .	25
3.1.	Tabla comparativa de las tecnologías disponibles en las bandas: 433 MHz, 868 MHz, 2.4 GHz y 5.6 GHz . . . . .	38
4.1.	Características técnicas de módulos de comunicación de alta potencia en la banda 868MHz. . . . .	44
4.2.	Posición y altura de los nodos en la implementación de los enlaces sim- ples de datos. . . . .	63
4.3.	Medidas de porcentaje de paquetes recibidos correctamente y RSSI rea- lizadas para varios enlaces sobre el mar. . . . .	64
5.1.	Comparación de protocolos de encaminamiento basados en el AODV .	88
5.2.	Tramas definidas en el sistema . . . . .	90
5.3.	Campos de la trama RREQ del protocolo AODV-LAB . . . . .	92
5.4.	Campos de la trama RREP del protocolo AODV-LAB . . . . .	93
5.5.	Campos de la trama DATA del protocolo AODV-LAB . . . . .	94
5.6.	Campos de la trama ACK del protocolo AODV-LAB . . . . .	95
5.7.	Campos de la trama PREQ del protocolo AODV-LAB . . . . .	95
5.8.	Campos de la trama PREP del protocolo AODV-LAB . . . . .	96



# Parte I

## Capítulos



# Capítulo 1

## Introducción

*Cuando hayáis talado el último árbol, matado el último animal y contaminado el último río, os daréis cuenta de que el dinero no se come.*

Proverbio de los indios Cris, Canadá

**RESUMEN:** En este capítulo se presenta la motivación, circunstancias y contexto en que se desarrolla este PFC (*Proyecto Fin de Carrera*). A continuación, se definen los objetivos generales y específicos, y se expone la metodología de trabajo y comunicación utilizados en su realización. Por último, se presenta la organización del presente documento, describiendo brevemente el contenido de cada capítulo que lo integra.

### 1.1. Motivación

El ruido en el mar ha existido siempre, de forma natural o biológica. Sin embargo, en los últimos cien años, la introducción masiva de fuentes sonoras artificiales en este medio ha crecido hasta una escala nunca experimentada. Por ello, debido a su carácter reciente y no controlado, el ruido antropogénico se ha convertido en una amenaza para su equilibrio. (André et al., 2009)

Las fuentes sonoras producidas por actividades humanas presentan efectos físicos, fisiológicos y de comportamiento sobre la fauna marina: mamíferos, reptiles, peces e invertebrados. Éstos pueden ser de diversa magnitud según la cercanía a la fuente sonora. Entre estos impactos se incluyen, por ejemplo:

- Una reducción de la abundancia de especies pesqueras hasta en un 50 % en las zonas de prospección. (Engås et al., 1993)(Skalski et al., 1992)
- Cambios de comportamiento y rutas migratorias de cetáceos. (Richardson et al., 1995) (Gordon y Mascrop, 1996)
- Daños de distinto rango, incluidos físicos, en vertebrados e invertebrados marinos. (McCauley et al., 2000) (Guerra et al., 2004)

Pueden existir, además, efectos a largo plazo a causa de una exposición crónica, y el sonido puede afectar a los animales indirectamente debido a cambios en la accesibilidad de sus presas, que sufren a su vez los efectos de la contaminación acústica. Estos daños

podrían afectar de forma significativa a la conservación de las poblaciones de especies de fauna marina amenazadas que utilicen áreas contaminadas acústicamente como ruta migratoria, área de reproducción o alimentación.

Las fuentes de contaminación acústica marina producidas por actividades humanas incluyen, entre otras: el transporte marítimo, la exploración y producción de gas y petróleo, el sonar militar e industrial, las fuentes de acústica experimental, las explosiones submarinas, militares o civiles, las actividades de ingeniería (dragados, construcciones, etc.), el ruido de aviones supersónicos, y la construcción y operación de parques eólicos.



(a) Ferry turístico de gran velocidad entre Las Palmas y Santa Cruz. (Eric Vandeville/©Rolex Awards)



(b) Plataforma petrolífera en el Mar del Norte. (Erik Christensen)



(c) Parque eólico marino de West of Duddon Sands, que contará con 108 aerogeneradores. (Paul Langrock Agentur)

Figura 1.1: Fuentes de contaminación acústica marina producidas por actividades humanas.

Estas fuentes sonoras se introducen en el espacio acústico y físico de los organismos marinos (ver Figura 1.2) y no existen actualmente niveles de referencia que permitan prever las consecuencias negativas de estas interacciones a corto, medio o largo plazo sobre el equilibrio natural de los océanos y sus efectos sobre la biodiversidad marina.

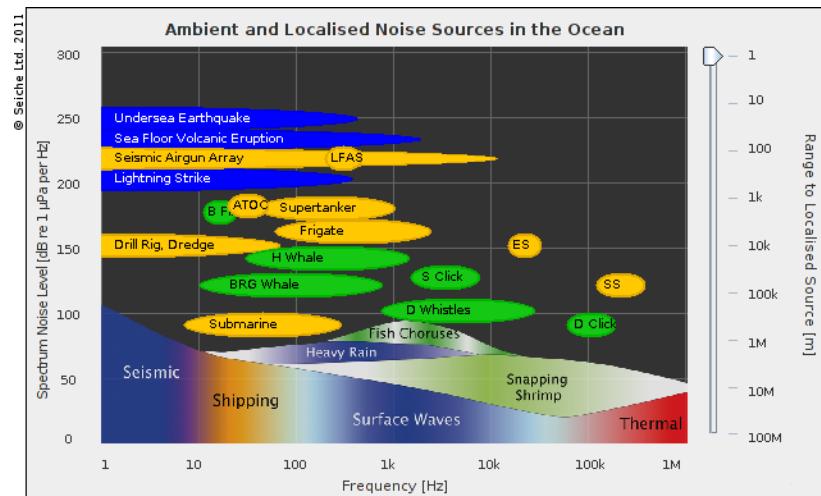


Figura 1.2: Niveles de sonido y frecuencias de fuentes sonoras antropogénicas y naturales del medio marino.

Evaluar el impacto acústico de fuentes sonoras artificiales en el medio marino es, en la actualidad, una tarea costosa, por varios motivos. El primero es la relativa falta de información sobre el mecanismo de proceso y análisis de sonidos por parte de los organismos marinos. Aunque es posible grabar y catalogar la mayoría de estas señales, no se conoce aun su papel e importancia en el equilibrio y desarrollo de las poblaciones. En segundo lugar, el posible impacto de emisiones sonoras no sólo concierne los

sistemas de recepción auditiva sino que puede intervenir a otros niveles sensoriales o sistémicos y resultar letal para el animal afectado. Si a estas dos razones de peso se añade el hecho que una exposición puntual o prolongada a un ruido determinado puede tener consecuencias negativas a medio y largo plazo y, por lo tanto, no observarse de inmediato, se entiende la gran dificultad a la cual se está confrontando la comunidad científica para obtener unos datos objetivos que permitan controlar de forma efectiva la introducción de ruido antropogénico en el mar.

El medio marino, como todo medio natural, se organiza sobre la base del equilibrio de los organismos que la habitan, cada uno de los cuales se sitúa a un nivel trófico específico que permite el desarrollo de los niveles superiores. Un desajuste de cualquiera de estos niveles desequilibra la cadena en los dos sentidos. Frente a un problema de conservación, el reto de los científicos es encontrar un organismo suficientemente representativo, es decir, cuyo equilibrio y desarrollo pueden influir de igual manera en el resto de la cadena trófica, y utilizarlo como bioindicador frente a la fuente contaminante.

La elección de los cetáceos como bioindicadores de la contaminación acústica del mar, no es fortuita. Ellos dependen del intercambio acústico para un gran número de actividades y comportamientos vitales como: la comunicación, orientación, relación con el medio, alimentación, y la gran mayoría de actividades dentro de su grupo social (cohesión, alarmas, relaciones materno-familiares, etc.). Por todo ello, los cetáceos representan, hasta la fecha, el mejor bioindicador de los efectos de la contaminación acústica en el mar.

Aunque el ruido ambiental en tierra fue regulado con anterioridad, la contaminación acústica marina se ha introducido en los marcos legales internacionales sólo recientemente<sup>1</sup>, traspasándose a las regulaciones nacionales de algunos países como el Reino Unido.

El Consejo Superior de la Sociedad Europea de Cetáceos (*European Cetacean Society*), sociedad que agrupa a los 500 científicos europeos que dedican sus investigaciones a la biología de los cetáceos, considera que<sup>2</sup>:

- Se necesita de forma urgente investigación sobre los efectos de la contaminación acústica humana en el mar, investigación que deberá conducirse bajo los más altos estándares de credibilidad científica, evitando los conflictos de intereses.
- Se deben desarrollar e implementar lo antes posible medidas de mitigación no intrusivas.
- Se tendría que limitar el uso de fuentes sonoras submarinas potentes hasta que se conozcan los efectos a corto, medio y largo plazo sobre los mamíferos marinos y evitar este uso en áreas de concentración de estas especies.
- Se deben desarrollar instrumentos legislativos en materia de contaminación acústica marina que permitan cumplir las políticas europeas y nacionales de protección de la biodiversidad marina. (André y Nachtigall, 2007)

Además del riesgo ambiental que supone el ruido antropogénico en el mar, existe otra serie de problemas asociados a la actividad humana en este medio, por ejemplo:

<sup>1</sup>Estas regulaciones incluyen los artículos 192, 194 (2,3), 206 y 235 de UNCLOS 1982 , y UNCED 1992

<sup>2</sup>Conclusiones del 17 Congreso Internacional de esta sociedad celebrado en Las Palmas de Gran Canaria en marzo de 2003 bajo el tema principal *Marine Mammals and Sound*

el *bycatch*, captura accidental de animales que no son objeto de pesca, o las colisiones entre barcos y cetáceos u otras especies.



(a) Existen varios estudios que coinciden en relacionar varamientos masivos con operaciones militares en las que se utilizaron señales de sonar de frecuencias medias y bajas. En la foto playa de Naracoopa, en la isla australiana de King. Marzo 2009. (Reuters)



(b) Captura accidental de la vaquita marina, una especie de marsopa, en el Golfo de California. (Cristian Faezi y Omar Vidal)



(c) Alrededor del 20% de las ballenas encontradas muertas en el Mediterráneo ha chocado con barcos. (©BBC News)



(d) Un cachalote muere tras una colisión con un ferry de alta velocidad que viajaba entre las islas de Gran Canaria y Tenerife. (Vidal Martin)

Figura 1.3: Ejemplos del impacto ambiental generado por el ser humano en el medio marino.

## 1.2. Marco de trabajo

### 1.2.1. Laboratorio de Aplicaciones Bioacústicas

La creación del LAB (*Laboratorio de Aplicaciones Bioacústicas*)<sup>3</sup> representa la respuesta madura de un grupo multidisciplinar de científicos a la continua degradación acústica del medio marino que está teniendo lugar en el Atlántico Norte y el Mediterráneo.

El LAB nace para responder con soluciones tecnológicas a la conservación del hábitat marino y prevenir su deterioro progresivo. Tiene como objetivo principal el desarrollo de aplicaciones capaces de limitar los efectos del ruido antropogénico en el mar y contribuir al desarrollo sostenible de las actividades marítimas. De este modo, se podrá dar respuesta a las exigencias planteadas al respecto por las instituciones locales, nacionales e internacionales y la sociedad en general a través de mecanismos de control de la contaminación acústica marina.

<sup>3</sup><http://www.lab.upc.es>

Si bien la monitorización y seguimiento de las actividades con impacto ambiental es una necesidad aceptada generalmente y una obligación legal para cualquier clase de plan o proyecto, en el campo de la contaminación acústica subacuática resulta de una importancia verdaderamente vital. Las carencias de investigación comentadas en la Sección 1.1 podrían ser paliadas a través del establecimiento de estos sistemas en las correspondientes Declaraciones de Impacto Ambiental o en las directrices de gestión de las Áreas Marinas Protegidas.

Por ello, la monitorización y seguimiento de actividades generadoras de contaminación acústica marina, se encuentra entre las principales líneas de investigación del LAB (André et al., 2008)(André, 2009). Como principal herramienta para ello se destaca la PAM (*Passive Acoustic Monitoring*, Monitorización mediante Acústica Pasiva). Grabar mediante hidrófonos las señales específicas que producen muchas especies y sub-grupos de ellas, permite revelar su presencia en zonas de interés de manera no intrusiva, es decir, sin añadir más ruido al medio. Como el sonido se propaga de manera muy eficiente en el agua, el alcance de la detección puede ser muy grande, superior a 100 Km. en condiciones favorables para las señales de baja frecuencia. Esto excede en gran medida las posibilidades de detección visual. Además, los avances en electrónica, informática y análisis numérico permiten ahora que esta técnica pueda aplicarse con una tecnología barata y accesible, convirtiéndola en una solución viable y sostenible para la mitigación de los efectos de la contaminación acústica marina.

Concretamente, el LAB desarrolla técnicas de DCL (*Detección, Clasificación y Localización*) de eventos acústicos y las implementa en tiempo real en diferentes soportes o sistemas integrados, como: observatorios submarinos cableados, boyas autónomas, estructuras sumergibles, planeadores, etc. (André et al., 2011) (Zaugg et al., 2010) (Houégnigan et al., 2010)

### 1.2.2. Proyecto LIDO

En las próximas dos décadas, se prevé un incremento de los niveles de desarrollo industrial en alta mar. Por ejemplo, los océanos, cada vez más, van apareciendo como una fuente alternativa de energía, gracias al desarrollo de parques eólicos marinos y de otras tecnologías para obtener energía de las olas o las mareas. Todo ésto, entre otros problemas, se traducirá en un aumento de los niveles de contaminación acústica en este medio, lo que, como se vio anteriormente, repercutirá gravemente en su equilibrio. Por ello, el problema que enfrenta actualmente la industria, y en general la sociedad, es que muchas actividades de importancia económica en el mar están en riesgo debido a la falta de información sobre los efectos del ruido antropogénico sobre los mamíferos marinos y, especialmente, la falta de herramientas disponibles para mitigar estos efectos. El desafío pues, es desarrollar soluciones tecnológicas que combinen los intereses de la industria y el buen estado ecológico de los océanos, logrando así un desarrollo sostenible.

El LAB está actualmente liderando un proyecto financiado por la Unión Europea en el marco de ESONET (*European Sea-floor Observatory Network of Excellence*)<sup>4</sup> llamado LIDO (*Listening to the Deep-Ocean Environment*)<sup>5</sup>. Se busca establecer una base científica centrada en cinco aspectos principales:

- Identificación y clasificación automática de fuentes de ruido artificiales, naturales y biológicas, para el posterior análisis de sus interacciones.

<sup>4</sup><http://www.esonet-noe.org>

<sup>5</sup><http://www.listentothedepth.com>

- Monitorización de los organismos marinos y su dinámica de poblaciones.
- Evaluación y control de los efectos a largo plazo de las fuentes antropogénicas de ruido en los organismos marinos, especialmente en zonas de seguridad o exclusión<sup>6</sup>.
- Educación social y de las administraciones.
- Aprobación de normas éticas y procedimientos objetivos y estandarizados en el desarrollo de actividades en alta mar en presencia de cetáceos, de una manera científicamente contrastada garantizando un desarrollo sostenible.

El software implementado en este proyecto está actualmente activo en los observatorios submarinos, conectados por cable a la costa, de: ANTARES (Colaboración Antares, Europa), OBSEA (UPC, Universidad Politécnica de Cataluña, España), Neptuno (University of Victoria, Canadá), Kushiro y Hatsushima (JAMSTEC, Japón) y NEMO (INFN, Italia). Además, parte del sistema está siendo probada para determinar su viabilidad en planeadores autónomos, en colaboración con el NURC.

En tiempo real, el software divide las grabaciones en diferentes bandas de frecuencias cubriendo los nichos acústicos de la mayor parte de especies de animales marinos. A continuación, cada banda es analizada por una serie de algoritmos organizados en módulos, concretamente de: detección, clasificación, localización de fuentes y medida de ruido. Con ello, se está creando una base de datos única y, además, está permitiendo crear sets de entrenamiento y calibración para el propio sistema, en función de: el estado del mar, la ubicación geográfica, el nivel de ruido, etc. En resumen el sistema LIDO DCL, toma como entrada una secuencia de datos acústicos y produce una salida con la caracterización de los eventos que se han detectado en ellos, espectrogramas para una rápida visualización y audio comprimido. Esta salida se pone a disposición del público en Internet, donde se puede ver con una aplicación específica (ver Figura 1.4). La interfaz pública se puede encontrar en la página web del proyecto.

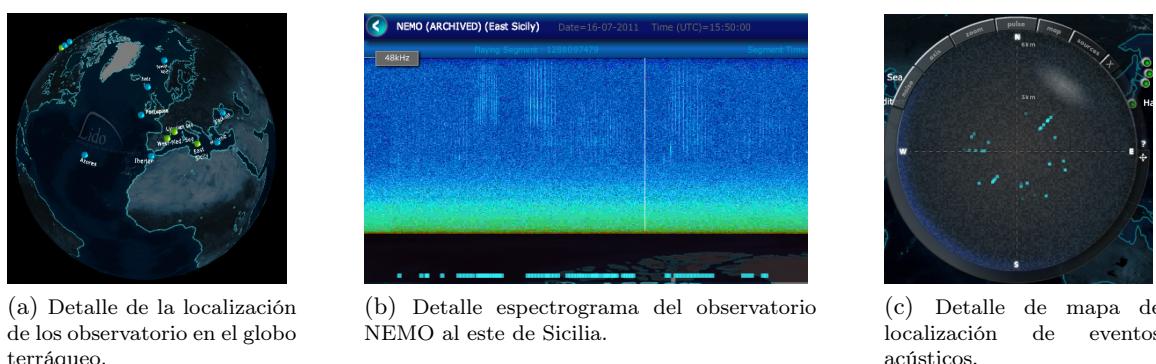


Figura 1.4: Detalles de la interfaz pública del proyecto LIDO.

Por su funcionalidad en tiempo real, y la posibilidad de integrarse en diferentes soportes, como: observatorios cableados, boyas autónomas, estructuras sumergibles, planeadores, etc; el sistema LIDO DCL, con la incorporación de un servicio de alarma, constituye una gran herramienta de mitigación no intrusiva.

<sup>6</sup>Entiéndense como tal, cualquier área marina protegida o cualquier otra área, de la cual, en virtud de su sistema de gestión o de una degradación de impacto ambiental se hayan excluido las posibilidades de desarrollar actividades que generen contaminación acústica submarina.

Actualmente, LIDO DCL permite el análisis exhaustivo de datos, sin restricción de frecuencias ni consumo, cuando el sistema está cableado. En su utilización en boyas autónomas, existen dos casos:

- Alimentación por paneles solares. Se acerca a las condiciones de análisis de los sistemas cableados, pero la transmisión de datos está limitada en alcance y velocidad por el radioenlace.
- Alimentación por baterías. Presenta las mismas limitaciones que el caso anterior en la transmisión de datos y, además, requiere algunas restricciones en el análisis de los mismos.

En el caso de desplegar varias boyas, todas ellas deberían tener conexión directa, punto a punto, con la estación base, formando una topología estrella (ver Figura 1.5a). Pero existe otra opción, permitir la intercomunicación entre boyas, que se convierten así en nodos de una red multisalto, con capacidad de enrutamiento (ver Figura 1.5b). De esta forma, basta con que una de ellas tenga conexión con la estación base para asegurar que todos los nodos de la red puedan comunicarse con ésta. Añadir dicha funcionalidad al sistema permitirá, entre otras cosas: eliminar la limitación de alcance comentada anteriormente, cubrir mayores áreas para DCL y mitigación, contrastar el análisis de las señales que se reciben desde boyas cercanas, etc. De esta manera, se obtendría una solución económica que permite la PAM con una resolución espacial, además de temporal. Los datos de todos los nodos se envían, casi en tiempo real, a una estación en tierra, o a una determinada embarcación, permitiendo así tomar medidas de mitigación a tiempo. Algunas aplicaciones, entre otras muchas, son:

- Creación de perímetros de seguridad entorno a fuentes sonoras de alta potencia, como pueden ser: barcos de prospección petrolífera, parques eólicos, actividades de ingeniería, explosiones y pruebas con sónar militar, experimentos científicos, etc.
- Detección de cetáceos, u otros animales, en áreas de pesca para evitar el bycatch.
- Alertar de la presencia de cetáceos en zonas de alto tráfico marítimo, para que los barcos puedan reducir su velocidad y prevenir colisiones, o en caso de producirse, minimizar sus consecuencias.

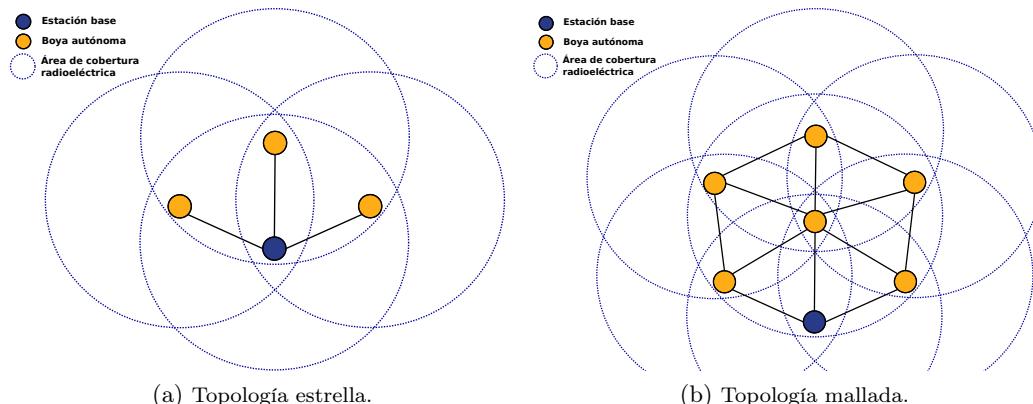


Figura 1.5: Ejemplos de topologías de red.

La tecnología utilizada para la implementación de esta red, determinará diferentes parámetros del sistema, como:

- La velocidad de transmisión de datos, que se traducirá en la capacidad de envío del flujo bruto de información, únicamente de alarmas o soluciones intermedias.
- El soporte de diversas topologías de red.
- La distancia máxima posible entre boyas, y a la estación base, que puede ser un barco, un receptor en tierra, una estación marina con conexión a tierra vía cable, etc.
- El consumo de energía.
- El tipo de boyas necesaria.
- Otros.

Por ello, partiendo de la problemática medioambiental descrita anteriormente, y del estado actual de desarrollo de las herramientas de mitigación para la misma, surgió el presente PFC, con la finalidad de establecer las bases y demostrar la viabilidad de la implementación del sistema LIDO DCL sobre una red multisalto de boyas autónomas. Este proyecto se ha desarrollado en el LAB, y está comprendido dentro del proyecto LIDO.

### 1.3. Objetivos

Se definen los siguientes objetivos para el presente PFC.

- **Objetivo General:** Diseñar y realizar un prototipo de red multisalto que operará en el medio marino con aplicación en la PAM de fuentes de ruido artificiales, naturales y biológicas.
- **Objetivos específicos:**
  - Realizar un estudio de posibilidades tecnológicas para la implementación de una red multisalto en el medio marino, donde cada nodo pueda encaminar la información para comunicarse con otros con los que no tiene conexión directa. El estudio debe tener en cuenta varios aspectos como: la distancia máxima de la conexión entre nodos en dicho medio, la velocidad de transmisión de datos, el consumo de energía, el precio de los equipos, etc.
  - Definir, basándose en los resultados del estudio anterior, el prototipo de red a implementar, incluyendo la tecnología que se utilizará y cómo se integrará con el sistema LIDO DCL ya desarrollado.
  - Realizar un cálculo teórico del alcance esperado para un radioenlace simple de datos en la frecuencia de operación seleccionada.
  - Implementar un radioenlace simple de datos en entorno marino.
  - Simular y verificar las funcionalidades esperadas a nivel de red.
  - Poner en marcha el prototipo de red con flujo real de datos.

## 1.4. Metodología de trabajo

Para la consecución de los objetivos del PFC, tal y como muestra la Figura 1.6, se ha utilizado una metodología de trabajo basada en cuatro líneas paralelas con diferente inicio temporal:

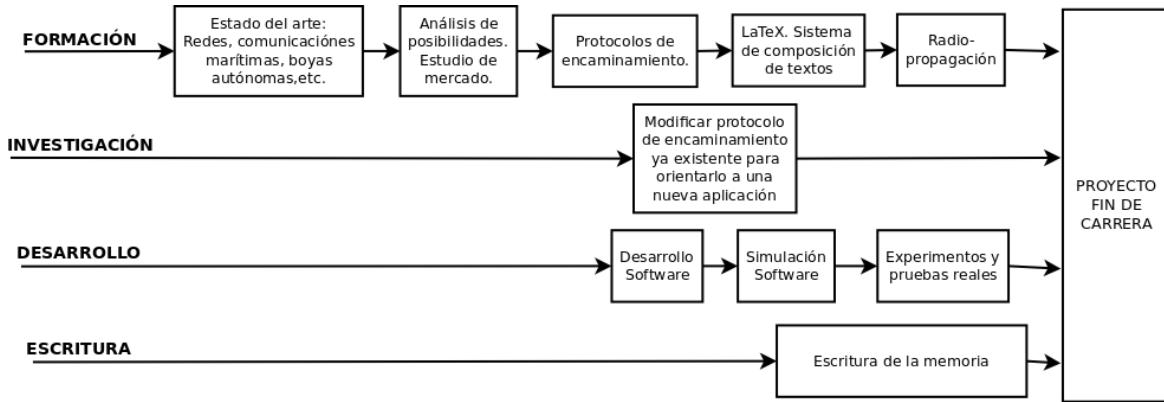


Figura 1.6: Diagrama de la metodología de trabajo.

- **Formación.** La lectura de publicaciones científicas, libros y otras fuentes, ha sido constante durante todo el desarrollo del PFC. Inicialmente se sintetizó el estado del arte en redes multisalto, radiocomunicaciones en el mar, boyas autónomas, etc. Tras el análisis de posibilidades y el estudio de mercado, como ya se verá más adelante, se centró el estudio en protocolos de encaminamiento. Finalmente, se repasó y completó los conocimientos en radiopropagación y balance de radioenlaces, para la realización de las pruebas de campo. En la etapa final, también se comenzó a trabajar con el sistema L<sup>A</sup>T<sub>E</sub>X, concretamente con la plantilla TEXIS, para la elaboración de la memoria.
- **Investigación.** Tras el estudio de varios protocolos de encaminamiento, y la necesidad detectada de tener que realizar la implementación de uno, se decidió realizar algunas modificaciones sobre el original, con el objetivo de lograr una solución más sencilla para dar soporte a una nueva aplicación.
- **Desarrollo.** En el análisis de mercado se detectó la necesidad de desarrollar una implementación propia de un protocolo de encaminamiento. Tras ello, se desarrolló también un sistema para la simulación y prueba del mismo. Finalmente, se procedió a la integración de todas las partes del sistema y la realización de la prueba final del prototipo, y a su evaluación.
- **Escritura.** La escritura se ha llevado a cabo a lo largo de todo el proyecto, condensando la mayor carga de trabajo al finalizar el periodo de desarrollo.

Además, durante todo el proceso de realización del PFC, se ha mantenido una comunicación periódica y un seguimiento del trabajo realizado con:

- Michel André, director del LAB, en los temas conceptuales y de definición de la herramienta final.

- Mike van der Schaar, responsable de la parte de software del sistema LIDO, en el desarrollo de código y simulación del mismo.
- Joan Vicent Castell, responsable de la parte de hardware del sistema LIDO, en la elección de los equipos de comunicación y realización de pruebas de campo.

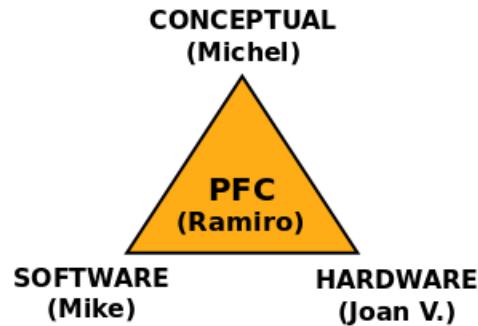


Figura 1.7: Comunicación con los compañeros del LAB.

## 1.5. Organización del documento

El presente documento está organizado de la siguiente forma:

- Capítulo 1: En él se presenta la motivación, circunstancias y contexto en que se desarrolla este PFC, sus objetivos y metodología de trabajo y comunicación.
- Capítulo 2: Se presenta todo el estudio del estado del arte que permite al lector ponerse en el punto de partida de este trabajo. Se incluye información necesaria para comprender las tecnologías de las que se hablará, se presenta la literatura científica consultada que tiene relación con el tema y que se va a utilizar más adelante como base de trabajo, y se presentan también las experiencias técnicas de otros investigadores en temas aplicables a este caso, y que serán de importancia a la hora de definir el prototipo a implementar.
- Capítulo 3: Se discutirá las posibles soluciones en base a toda la información recopilada en el capítulo anterior y se definirá el prototipo de red a implementar, incluyendo la tecnología que se utilizará y cómo se integrará con el sistema LIDO DCL ya desarrollado.
- Capítulo 4: Se tratará en profundidad la parte hardware del sistema. Estudio de mercado, módulos de comunicación seleccionados, configuración de los mismos y realización de un enlace simple de datos con ellos en entorno marino.
- Capítulo 5: Se hablará en detalle de la parte software del sistema. Estudio de protocolos de encaminamiento, definición de la solución a implementar, descripción del desarrollo del software, presentación del método de simulación utilizado y validación de funcionalidades esperadas.
- Capítulo 6: Descripción de las pruebas finales del prototipo de red, integrando las partes de hardware y software. Resultados obtenidos.
- Capítulo 7: Conclusiones que se pueden extraer de los resultados obtenidos en la realización del PFC y contraste con los objetivos fijados inicialmente. Se plantean también las líneas de investigación y desarrollo futuras que se abren a raíz de los resultados y conclusiones de este trabajo.



## Capítulo 2

# Fundamentos y estado del arte

*Aprende como si fueras a vivir toda la vida,  
y vive como si fueras a morir mañana.*

Charles Chaplin

**RESUMEN:** En este capítulo, en primer lugar, se realiza un estudio de las bandas de frecuencia en las que habrá posibilidad de operar y las limitaciones legales de cada una. A continuación, se presentan varios tipos de redes multisalto, concluyendo que en este proyecto se implementará una de tipo ad hoc por sus características. También se realiza un análisis de los factores que afectarán a un radioenlace en entorno marino. Por último, se hace una breve introducción de las tecnologías Wi-Fi, WiMAX y ZigBee, por ser las más presentes en el mercado con aplicación en redes multisalto, y se exponen algunas experiencias técnicas de otros investigadores en la implementación de enlaces simples o redes en el medio marino.

### 2.1. Bandas de frecuencia

El espectro radioeléctrico, en primer lugar, se divide en bandas de frecuencia, que son asignadas a los diferentes servicios o sistemas radioeléctricos por la UIT (*Unión Internacional de Telecomunicaciones*). En segundo lugar, las administraciones de los diferentes países efectúan las atribuciones de frecuencias. Por ello, la utilización de sistemas radioeléctricos debe hacerse dentro del régimen de legalidad vigente, permitiendo la normal coexistencia de las comunicaciones de esta naturaleza.

Las bandas libres, o de uso común, dentro del espectro radioeléctrico, son aquellas en las que se permite la operación de dispositivos de radiocomunicaciones sin una autorización individual de cada estación. Por otra parte, se tiene las bandas con licencia, en las que se asigna una frecuencia o canal para un uso exclusivo del mismo. Obtener una licencia de operación en estas bandas supone la realización de ciertos trámites burocráticos, y la inversión de tiempo y dinero. Por ello, y por la limitación que supone tener que ajustarse en cada país o región a sus restricciones y condiciones para obtener la correspondiente licencia, para este proyecto se estudiarán únicamente las bandas libres.

Esta terminología de clasificación es muy utilizada en la jerga del sector, pero en los textos de regulación del espectro no se emplea como tal, y se habla de las bandas

para aplicaciones ICM (*Industriales, Científicas y Médicas*). Éstas fueron definidas por la UIT en el artículo 5 del RR (*Reglamento de Radiocomunicaciones*)<sup>1</sup>. Las bandas ICM están reservadas internacionalmente<sup>2</sup> para el uso no comercial de radiofrecuencia electromagnética en las áreas industrial, científica y médica. La utilización de estas bandas de frecuencia está abierto a todo el mundo sin necesidad de licencia, respetando las regulaciones que limitan los niveles de potencia, ancho de banda, etc. Este hecho a dado lugar a su uso en comunicaciones (p.e. WLAN o WPAN), donde la tecnología empleada cuenta con una cierta tolerancia a errores y utiliza mecanismos de protección contra interferencias, como técnicas de espectro ensanchado<sup>3</sup>, y de acceso múltiple a la banda.

Concretamente, las bandas ICM actuales se muestran en la Tabla 2.1:

Banda	Frecuencia central	Unidad	RR No.	Disponibilidad
6.765 - 6.795	6.780	MHz	5.138	Sujeta a la aceptación local
13.553 - 13.567	13.560	MHz	5.150	
26.957 - 27.283	27.120	MHz	5.150	
40.66 - 40.70	40.680	MHz	5.150	
433.05 - 434.79	433.92	MHz	5.138	Región 1 y sujeta a la aceptación local. Ver RR No. 5.280
902 - 928	915	MHz	5.150	Únicamente Región 2
2.4 - 2.5	2.45	GHz	5.150	
5.725 - 5.875	5.8	GHz	5.150	
24 - 24.25	24.125	GHz	5.150	
61 - 61.5	61.25	GHz	5.138	Sujeta a la aceptación local
122 - 123	122.5	GHz	5.138	Sujeta a la aceptación local
244 - 246	245	GHz	5.138	Sujeta a la aceptación local

Tabla 2.1: Bandas ICM definidas por la UIT.

En la Figura 2.1 se muestra un mapa con las regiones definidas por la UIT. Además, se destaca que, como se observa en la Tabla 2.1, las bandas definidas en RR No. 5.138 tienen su uso sujeto a una autorización especial de la administración en cuestión, de acuerdo con las otras administraciones cuyos servicios de radiocomunicación puedan resultar afectados. Por último, para todas las bandas ICM la UIT advierte: *Los servicios de radiocomunicación que operen en estas bandas deben aceptar la interferencia perjudicial que puede ser causada por las aplicaciones ICM.*

En Europa, la banda 902 - 928 MHz se utiliza para aplicaciones de telefonía móvil automática GSM, por ello, se ha definido<sup>4</sup> otra banda de 868 a 870 MHz para dispositivos de corto alcance, SRD (*Short Range Devices*). Ésta se encuentra dividida en sub-bandas destinadas a diferentes aplicaciones.

En España, las restricciones de potencia y condiciones de utilización aplicadas

<sup>1</sup>Concretamente en los puntos 5.138 y 5.150.

<sup>2</sup>Existen algunas excepciones a nivel regional.

<sup>3</sup>Ver RR 15.13.

<sup>4</sup>ETSI TR 102 649-2 V1.1.1 (2008-09).

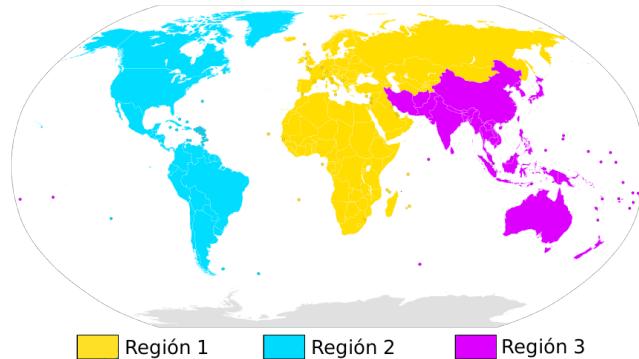


Figura 2.1: Regiones definidas por la UIT en su Reglamento de Radiocomunicaciones.

sobre las bandas ICM definidas por la UIT se recogen en el CNAF (*Cuadro Nacional de Atribución de Frecuencias*), publicado por el Ministerio de Industria, Turismo y Comercio. En la Tabla 2.2, se recogen los valores actualmente en vigor según las Notas de Utilización Nacional del CNAF 2010.

A la hora de optar por trabajar en una u otra banda de frecuencia deberá tenerse en cuenta:

- La existencia de normas o estándares de operación en la banda que proporcionen criterios uniformes en el ámbito territorial más extenso posible, de manera que se pueda garantizar la interoperabilidad a nivel global. Los estándares consensuados, o *de iure*, suelen contar con un respaldo más amplio y con menos detractores que los estándares *de facto*, puesto que éstos últimos pueden contener tecnologías privativas protegidas con patentes y para las que no negocien acuerdos de licencia, algo que no es habitual en los estándares elaborados por organismos.
- La presencia en el mercado de tecnologías que operen en dicha banda. Cuanto mayor sea, es decir, cuanto mayor oferta haya, más se abaratará esa tecnología. Además, su disponibilidad también será mayor.
- La potencia de transmisión y sensibilidad de los módulos de comunicación. Cuanto más elevada sea la diferencia entre estos dos parámetros, mayor será la distancia posible entre los nodos de la red.
- La frecuencia de operación. Las frecuencias altas permiten una velocidad de transmisión de datos más elevada, pero se ven más afectadas por fenómenos como: la difracción, la refracción, los agentes meteorológicos, etc. Ésto se traduce en una reducción del alcance de la señal e incluso, en algunos casos, en una pérdida temporal de la comunicación.(McLarnon, 2001)
- La modulación, el protocolo de acceso al medio y otros parámetros que también influyen en: la calidad, el alcance y la robustez del sistema de comunicación.

<sup>5</sup>Técnicas de Control de Potencia.

<sup>6</sup>Redes de Área Local.

Banda	Unidad	Campo magnético / Potencia máxima	Nota	Observaciones
6.765 - 6.795	MHz	42 dB $\mu$ A/m (a 10 m)	UN-115	Aplicaciones de bucle inductivo
13.553 - 13.567	MHz	42 dB $\mu$ A/m (a 10 m)	UN-115	Aplicaciones de bucle inductivo
26.957 - 27.283	MHz	10 mW pra	UN-4	Aplicaciones genéricas de SRD. Coincide con los primeros 28 canales de la Banda Ciudadana, CB-27
40.66 - 40.70	MHz	10 mW pra	UN-11	Aplicaciones genéricas de SRD
433.05 - 434.79	MHz	1 mW pra 10 mW pra	UN-30	Ciclo de trabajo 100 % Ciclo de trabajo 10 %
434.04 - 434.79	MHz	10 mW pra	UN-30	Ciclo de trabajo 100 % Canalización $\leq 25\text{kHz}$
869.40 - 869.65	MHz	500 mW pra	UN-39	C. trabajo 100 % con canalización $25\text{kHz}$ C. trabajo 10 % sin canalización
2.4000 - 2.4835	GHz	100 mW pire	UN-85a	La norma de referencia para los equipos es el estándar ETSI EN 300 328 en su versión actualizada
5.470 - 5.725	GHz	1 W pire con TPC <sup>5</sup> 500 mW pire sin TPC	UN-128	No pertenece a las Bandas ICM, pero en España es de uso común para RLANS <sup>6</sup>
5.725 - 5.875	GHz	25 mW pire	UN-130	La norma técnica de referencia para estos dispositivos es la norma ETSI EN 300 440
24 - 24.25	GHz	100 mW pire	UN-86	Aplicaciones no específicas de baja potencia
61 - 61.5	GHz	100 mW pire	UN-115	Aplicaciones no específicas de baja potencia
122 - 123	GHz	100 mW pire	UN-115	Aplicaciones no específicas de baja potencia
244 - 246	GHz	100 mW pire	UN-115	Aplicaciones no específicas de baja potencia

Tabla 2.2: Notas de Utilización Nacional de las Bandas ICM definidas en el CNAF 2010.

## 2.2. Redes ad hoc

Las redes ad hoc (Remondo y Niemegeers, 2003) están constituidas por dispositivos inalámbricos que se comunican mediante radioenlaces sin la necesidad de una infraestructura de red preexistente, como routers en redes cableadas o puntos de acceso, AP (*Access Point*), en redes inalámbricas con gestión centralizada. Las principales características de estas redes son:

- **Autoconfiguración.** No se requiere un sistema central para la configuración y gestión de la red. Esta es capaz, de manera descentralizada, de descubrir nuevos nodos y eliminar aquellos que ya no se encuentran presentes o en funcionamiento.
- **Capacidad de enrutamiento de todos, o casi todos, los nodos de la red.** Esto permite la comunicación entre aquellos que se encuentran fuera del alcance directo de la señal radio. Esta característica se conoce con el nombre de capacidad multisalto, ya que un determinado paquete puede pasar por varios nodos intermedios antes de llegar a su destino. Además, existe una redundancia de rutas de modo que en caso de fallo de una, se buscará otra alternativa. Todo esto supone una mejor escalabilidad de este tipo de redes en comparación con las redes inalámbricas tradicionales.

- **Topología dinámica.** No se requiere una estructura fija, la red puede estar en constante cambio. Esto permite la posibilidad de tener nodos en movimiento dentro de la misma. No es necesaria la existencia de estaciones base. Además, es tolerante a fallos, puesto que la caída de un solo nodo no implica la caída del resto de la red, como puede ocurrir en topologías de árbol o cluster.

Las redes ad hoc realizan ciertas tareas de enrutamiento con el fin de optimizar su funcionamiento y calidad de servicio. En este ámbito existen actualmente abiertas varias líneas de investigación muy activas, debido a las grandes posibilidades que ofrecen este tipo de redes.

También se debe atender a las dificultades que habrá que enfrentar. En las comunicaciones inalámbricas, ocurre que el *throughput*<sup>7</sup> es mucho menor que la tasa máxima de transmisión radio. Esto es debido a diversos factores como: la contienda en el acceso múltiple, el fading, ruido, interferencias y otros. Además, como se vio en la Sección 2.1, existen bandas con limitación en el ciclo de trabajo. Por último, se destaca que las redes ad hoc presentan restricciones de ancho de banda debidas a la capacidad variable de sus enlaces inalámbricos.

En conclusión, las redes ad hoc son capaces de establecer una comunicación espontánea entre terminales fijos o móviles siempre que haya una posibilidad física para lograrlo. Esto da lugar a un amplio abanico de aplicaciones que pueden ser potencialmente soportadas por este tipo de redes, pero también suponen una serie de dificultades que habrá que superar.

### 2.3. Redes de malla inalámbricas

Conocidas por el acrónimo en inglés WMN (*Wireless Mesh Network*). A menudo, existe una confusión entre este tipo de redes y las ad hoc. Concretamente, las WMN pueden considerarse como un tipo especial de redes ad hoc inalámbricas. Básicamente son redes con topología de infraestructura, pero que permiten unirse a la red a dispositivos que, a pesar de estar fuera del rango de cobertura de los AP o nodos, están dentro del rango de algún otro dispositivo de la red que, directa o indirectamente, esté conectado a ella. Por otro lado, las redes ad hoc dependen únicamente de sus dispositivos y de sus rangos de cobertura, sin la utilización de AP. Otras diferencias destacables son:

- Los router en las WMN no tienen restricciones de recursos, lo que aumenta las posibilidades de la red. Esto no ocurre con los nodos de las redes ad hoc, que generalmente cuentan con unos recursos limitados cuya utilización debe optimizarse.
- Los nodos en las redes ad hoc son los encargados del enrutamiento y la configuración del sistema, mientras que las WMN contienen toda una red de routers que se encargarán de la conexión, además de la integración con otro tipo de redes ya existentes.
- Las WMN tienen una topología relativamente estable, excepto por fallos ocasionales de alguno de sus nodos, o por la entrada en la red de alguno nuevo. En las redes ad hoc, como se expuso en la Sección 2.2 la topología es dinámica.

<sup>7</sup>Volumen de información que fluye en las redes de datos.

Debido a que este proyecto está enfocado para su aplicación en el mar, donde no habrá ninguna infraestructura de comunicación previamente instalada, el prototipo a diseñar será de una red ad hoc, y así se le hará referencia a lo largo del documento.

## 2.4. Redes de sensores inalámbricas

Una red de sensores inalámbrica, en inglés WSN (*Wireless Sensor Network*), es una red de dispositivos autónomos, denominados nodos, distribuidos de forma espacial que utilizan sensores para monitorizar condiciones físicas y ambientales. Los nodos de medida se comunican de manera inalámbrica con un *gateway*, o nodo puente, el cual proporciona una conexión al entorno cableado donde se puede: adquirir, analizar y presentar los datos medidos. La conexión con el gateway, como se observa en la Figura 2.2, puede realizarse bien combinando estos dispositivos autónomos con varios routers, o bien formando una red ad hoc en la que cada nodo puede: actuar como emisor y receptor, ofrecer servicios de encaminamiento entre nodos sin visión directa, y registrar datos referentes a los sensores locales del mismo. En algunos casos también interesa que los nodos no sólo puedan observar o tomar datos sino que además sean capaces de actuar para activar funciones de otros sistemas. Existen múltiples aplicaciones para las WSN en distintos campos tales como: entornos industriales, domótica, entornos militares, detección ambiental, eficiencia energética, medicina, automoción, etc.

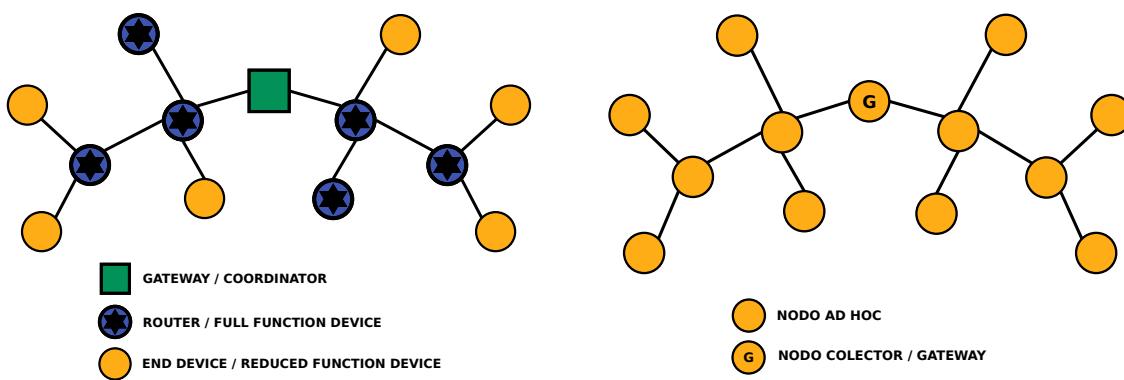


Figura 2.2: Posibles implementaciones de una red de sensores inalámbrica (WSN).

La arquitectura de un nodo se divide en:

- **Subsistema sensorial.** A través de sensores mide parámetros de diversa naturaleza (físicos, químicos, etc.) del medio ambiente.
- **Subsistema computacional.** Procesa la información analógica recogida por los diferentes sensores y la digitaliza para su envío.
- **Subsistema de comunicación.** Se encarga del intercambio de mensajes con los nodos vecinos, con el objetivo de hacer llegar la información al punto que interese, generalmente un *gateway*, o nodo colector.

Las redes de sensores tienen una serie de características propias y otras adaptadas de las redes ad hoc:

- **Topología dinámica.** En una red de sensores, a menudo la topología es cambiante y los nodos tienen que adaptarse para poder comunicar nuevos datos adquiridos.
- **Variabilidad del canal.** El canal radio es muy variable y en él existen una serie de fenómenos como pueden ser: la atenuación, desvanecimientos rápidos, desvanecimientos lentos e interferencias, que pueden producir errores en los datos.
- **No se utiliza infraestructura de red.** Una red sensora no tiene necesidad alguna de infraestructura para poder operar, ya que sus nodos pueden actuar de emisores, receptores o enruteadores de la información. Sin embargo, hay que destacar en el concepto de red sensora la figura del nodo colector, que es el que reúne la información de los demás. Esta información generalmente es adquirida por un ordenador conectado a este nodo y que a su vez podrá estar conectado a una base de datos, Internet, etc.
- **Comunicaciones multisalto o broadcast.** En aplicaciones sensoras siempre es característico el uso de algún protocolo que permita comunicaciones multisalto, por ejemplo: AODV, DSDV, OLSR u otros, aunque también es muy común utilizar mensajería basada en broadcast.
- **Consumo energético.** Es uno de los factores más importante debido a que tiene que, con una unidad de energía limitada, lograr un compromiso entre autonomía y capacidad de procesamiento. Un nodo sensor tiene que contar con un procesador de consumo ultra bajo así como de un transceptor radio y un software con la misma característica.
- **Limitaciones hardware.** Para poder conseguir un consumo ajustado, se hace indispensable que el hardware sea lo más sencillo posible, así como su transceptor radio, lo que se traduce en una capacidad de procesamiento limitada.
- **Costes de producción.** Dado que por su naturaleza una red de sensores suele contar con un número elevado de nodos para poder obtener datos con fiabilidad, los nodos sensores una vez definida su aplicación, deben ser económicos de fabricar en grandes cantidades.

Como se verá con mayor detalle en el siguiente capítulo, varias de estas características no se cumplirán en los sistemas de PAM (*Passive Acoustic Monitoring*, Monitorización mediante Acústica Pasiva). Por ello, aunque en gran parte el concepto de WSN si es aplicable a la PAM, en este proyecto no se hablará como tal de una red de sensores inalámbrica, sino de una red ad hoc con aplicación en la monitorización mediante acústica pasiva.

## 2.5. Radioenlaces en entornos marinos

### 2.5.1. Fundamentos básicos de propagación

La UIT-R define la propagación en espacio libre como la propagación de una onda electromagnética en un medio dieléctrico ideal homogéneo e isótropo que se puede considerar infinito en todas las direcciones. Este es un modelo ideal, que no existe en la realidad, si bien permite conocer las mínimas pérdidas que existen en la propagación.

La expresión para las pérdidas de propagación de ondas radioeléctricas en espacio libre en función de la distancia y la frecuencia de operación es la siguiente:

$$L_{bf}[dB] = 32,45 + 20 \cdot \log(d[km]) + 20 \cdot \log(f[MHz]) \quad (2.1)$$

, donde  $d[km]$  es la distancia en kilómetros entre el emisor y el receptor, y  $f[MHz]$  es la frecuencia de operación en megahercios.

En la realidad, el espacio se encuentra ocupado por obstáculos, como puede ser la misma superficie sobre la que se encuentran el emisor y el receptor. Al transmitir una onda, ésta puede incidir sobre dichos obstáculos produciendo reflexiones, de modo que el receptor además de recibir un rayo directo puede recibir otros reflejados. La combinación de estos ocasiona interferencias, que serán constructivas o destructivas según sea la fase y la amplitud de los diferentes rayos recibidos. El resultado de esta combinación es muy difícil de predecir, ya que depende de la posición exacta de todos los elementos que rodean la línea directa de transmisión. En el Capítulo 4, se profundizará más en este tema, presentando el modelo de propagación de tierra plana, o de dos rayos, para el estudio previo del enlace sobre la superficie del mar.

Siendo la fase mínima, tomando su valor como cero, la del rayo que une en línea recta el emisor y el receptor, se define la primera zona de Fresnel como el volumen de espacio entre estos dos puntos de modo que el desfase de las ondas reflejadas en dicho volumen no supere los  $180^\circ$ , contribuyendo destructivamente al rayo directo. Concretamente, este volumen tiene la forma de un elipsoide de revolución, como se puede observar en la Figura 2.3. La segunda zona abarca hasta un desfase de  $360^\circ$ , y es un segundo elipsoide que contiene al primero. Del mismo modo se obtienen las zonas superiores.

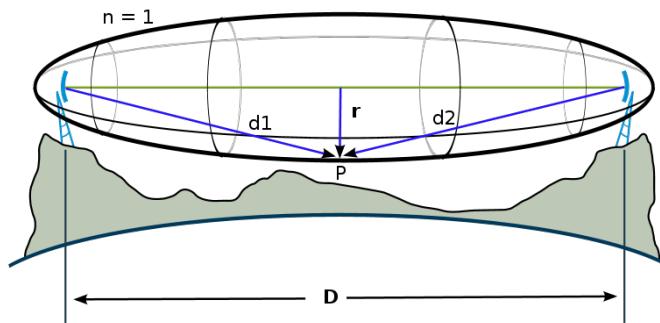


Figura 2.3: Zona de Fresnel.

La obstrucción máxima permisible para considerar un enlace con línea de visión, o LOS (*Line of Sight*), es el 40 % de la primera zona de Fresnel. Si el punto de reflexión se encuentra fuera de esta primera zona, la combinación de los rayos puede ser constructiva o destructiva dependiendo de la diferencia de fase. Si se sitúa exactamente en el límite, la combinación será constructiva. Por último, si el punto de reflexión se encuentra dentro de la primera zona de Fresnel, el enlace se considerará sin línea de visión, o NLOS (*Non Line of Sight*), y la combinación del rayo reflejado contribuirá siempre destructivamente.

La fórmula genérica de cálculo de las zonas de Fresnel es:

$$r_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \quad (2.2)$$

, donde:

- $r_n$ : radio de la enésima zona de Fresnel en metros ( $n = 1, 2, 3\dots$ ).
- $d_1$ : distancia desde el transmisor al objeto en metros.
- $d_2$ : distancia desde el objeto al receptor en metros.
- $\lambda$ : longitud de onda de la señal transmitida en metros.

Por lo tanto, el radio de la primera zona de Fresnel,  $r_1$ , conocida la distancia entre las dos antenas ( $d$ ) y la frecuencia de operación ( $f$ ), suponiendo al objeto en el punto central y en unidades del sistema internacional será:

$$r_1 = 17,32 \sqrt{\frac{d}{4 \cdot f}} \quad (2.3)$$

, donde:

- $r_1$ : radio en metros.
- $d$ : distancia en metros ( $d_1 = d_2, d = d_1 + d_2$ ).
- $f$ : frecuencia de operación en megahercios (MHz) ( $\lambda = \frac{c}{f}$ )

### 2.5.2. El entorno marino

En la nota de aplicación sobre *radioenlaces en entorno marino* elaborada por Albentia-Systems (2008), se realiza un estudio de los distintos fenómenos que afectan a la radiocomunicación en este medio. De él se destacan las siguientes conclusiones:

- La comunicación será NLOS, es decir, la reflexiones sobre la superficie del mar contribuirán destrutivamente a la señal recibida. Para la PAM en el medio marino interesa que la distancia entre los nodos sea del orden de unidades o decenas de kilómetros. Si en las boyas, la altura de las antenas sobre la línea de flotación está en torno a los 2 o 3 metros, la primera zona de Fresnel tocará directamente la superficie del agua. Como se dijo anteriormente, en el Capítulo 4 se evaluará la magnitud de este hecho utilizando el modelo de propagación de tierra plana.
- Cuanto más conductora sea la superficie de reflexión, más potente será el rayo reflejado. Por ello, la reflexión sobre el agua del mar, que al ser salada es más conductora, generará interferencias destructivas más fuertes que las que ocurrirán en un escenario idéntico pero con reflexiones sobre la superficie terrestre.
- La superficie marina es dinámica debido al oleaje y al efecto de las mareas que, continua y cíclicamente, suben y bajan. Esto provoca que la altura relativa de las antenas en un radioenlace sobre el mar esté continuamente cambiando y, por lo tanto, el punto, o los puntos, en los que la onda se refleja en la superficie cambian constantemente, provocando un efecto destructivo de mayor o menor magnitud.

- La presencia de gases (agua y oxígeno) e hidrometeoros (niebla, lluvia, nieve...) en la troposfera afectan a la comunicación atenuando la señal. Este efecto, sobre todo a frecuencias altas, no es un factor crítico, pero debe tenerse en cuenta ya que cuanto más próximo a la superficie del mar discurra el rayo, mayor será la humedad y la concentración de partículas en suspensión que atenúen la señal, como pequeñas gotas, salitre, etc.
- Un radioenlace sobre el agua presentas los mismos problemas para cualquier tecnología o banda de frecuencia de operación, aunque se darán en mayor o menor medida.

Yapicioglu y Oktug (2009) realizan un estudio sobre el efecto de la altura de las olas en la conectividad de una red de sensores implementada sobre la superficie marina. En sus simulaciones observan que este factor tiene un efecto muy notorio en el correcto funcionamiento de la red. Para solventar este efecto proponen la utilización de más nodos más próximos entre si, aunque éstos solo tengan función de enrutación o repetidor, sin el subsistema de sensores, reduciendo así costes y no añadiendo más tráfico innecesario a la red.

Existen más estudios, como (Palazzi et al., 2005), (Radha et al., 2005) y (Pérez et al., 2006), en los que se habla de otros efectos asociados a la propagación en un entorno marino como:

- Conductos por evaporación de agua. Debido a la evaporación del agua se forman zonas, o conductos, con un índice de refracción menor. Ésto puede provocar que las ondas viajen por su interior describiendo una curva hacia la superficie terrestre en donde se reflejan, dando lugar a un efecto de guía de onda que puede hacer que se alcancen distancias de propagación mucho mayores con respecto a los mecanismos de propagación normal de la onda. Estos conductos por evaporación pueden manifestarse durante días.
- Brisa marina. Este fenómeno varía las condiciones de refractividad de la atmósfera, lo que conlleva una modificación del trayecto de las ondas propagadas y, por lo tanto, afecta a la calidad de los radioenlaces. La brisa marina es un fenómeno local, su patrón de intensidad es diferente en cada localización dependiendo de factores como: la latitud, la temperatura, la topografía, etc.

## 2.6. Wi-Fi

Wi-Fi es una marca de la Wi-Fi Alliance, la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares IEEE 802.11 relacionados a redes inalámbricas de área local. El objetivo de la misma fue crear una marca que permitiese fomentar más fácilmente la tecnología inalámbrica y asegurar la compatibilidad de equipos. Actualmente, Wi-Fi es la tecnología inalámbrica por excelencia, a nivel internacional, en el mercado. Ésto la convierte en una opción económica y con bastante soporte.

Existen diversos tipos de Wi-Fi, basado cada uno de ellos en un estándar IEEE 802.11 aprobado. En la Tabla 2.3 se muestra un resumen de ellos:

En lo referente a la utilización de esta tecnología para la implementación de redes ad hoc, existe una gran variedad de protocolos de enrutamiento y software libre integrable con el sistema operativo GNU/Linux. Por ello, en caso de optar por esta



Figura 2.4: Logo Wi-Fi.

Estándar IEEE	Banda de frecuencia de operación (GHz)	Velocidad teórica (Mbps)	Velocidad real (Mbps)	Comentarios
802.11a	5.6	54	20	Banda menos utilizada que la de 2.4 GHz. Menos interferencias. A mayor frecuencia la absorción de las ondas es mayor y la presencia de obstáculos afecta más a la comunicación.
802.11b	2.4	11	5.9 sobre TCP 7.1 sobre UDP	Utiliza el método de acceso CSMA/CA, definido en el estándar original. El espacio ocupado por la codificación de este protocolo hace que la velocidad teórica se vea reducida en la realidad.
802.11g	2.4	54	22	Puede operar con sistemas basados en 802.11b, aunque, en ese caso, sus prestaciones empeoran.
802.11n	2.4 y 5.6	600	300	Compatible con el resto de sistemas basados en 802.11a, 802.11b y 802.11g.

Tabla 2.3: Tipos de Wi-Fi.

tecnología, estos desarrollos previos podrán utilizarse libremente y, en caso de ser necesario, modificarse apropiadamente según sea de interés. También existen soluciones en las que se integra en un único producto el sistema de radio y el de red, aunque en la mayoría de los casos son de carácter privativo, es decir, sin posibilidad de disponer del código para su modificación.

A continuación, se presentan algunos casos reales en los que se ha utilizado esta tecnología, tanto para implementar enlaces simples como para redes ad hoc o de malla, donde la distancia entre las estaciones es del orden de kilómetros:

- La fundación EHAS (*Enlace Hispano Americano de Salud*) promueve el uso apropiado de las nuevas TIC (*Tecnologías de la Información y la Comunicación*) para mejorar los procesos de salud en zonas rurales aisladas de países en desarrollo. Una de sus principales líneas de investigación es la utilización de la tecnología Wi-Fi (802.11b y 802.11g), por su reducido coste y madurez, para la implementación de redes ad hoc, o redes de malla estáticas. Concretamente, estudian la aplicación de esta tecnología para enlaces de larga distancia, trabajando en zonas donde la regulación existente permite operar con una potencia más elevada de la definida por la UIT para la banda ICM de 2.4 GHz. Además, ajustan los timeouts y otros parámetros para optimizar su funcionamiento para estas distancias. Entre sus resultados publicados se destacan enlaces punto a punto con antenas direccionales de alta ganancia (19 - 24 dBi) de 42 kilómetros con un throughput de 2.5 Mbps trabajando con 802.11g a 6 Mbps, y 84 kilómetros con un throughput de 1.5 Mbps con 802.11b a una velocidad de 2 Mbps. Con antenas omnidireccionales de alta ganancia (12 - 15 dBi) han logrado enlaces de hasta 5 kilómetros. En todos los casos, tanto el emisor como el receptor se encuentran situados estáticamente a una altura considerable, logrando mantener la primera zona de Fresnel despejada y, por lo tanto, tratándose de enlaces LOS. Además, utilizan un sistema operativo GNU/Linux y protocolos de encaminamiento de código abierto. (Simó et al., 2005) (Simó et al., 2006a) (Simó et al., 2006b)
- Existen algunas empresas, como *Wi-Fi for Boats*<sup>8</sup> o *The Wirie*<sup>9</sup>, que comercializan soluciones Wi-Fi de larga distancia para suministrar acceso a Internet a embarcaciones próximas a la costa. Llama la atención que en la mayoría de los casos el equipo utilizado en el barco consta del AP *Bullet2* de la compañía *Ubiquiti Networks*<sup>10</sup> junto con antenas omnidireccionales de alta ganancia, 15 dBi generalmente. Los resultados obtenidos, según las experiencias que narran los usuarios en diversos foros, varían entre alcances de 1 kilómetro escaso hasta casi 5. Esto dependerá en gran medida también del equipo disponible en la costa, del que no se da ningún dato.

Por otra parte, cabe destacar que varios productos de *Ubiquiti Networks* también son altamente utilizados por los integrantes de redes libres, como *Guifi*<sup>11</sup>. Esto se debe a sus prestaciones radio, su facilidad de utilización y montaje, y a la posibilidad de utilizarse para formar un WDS (*Wireless Distribution System*, Sistema de Distribución Inalámbrico), lo que permite la interconexión inalámbrica de APs en una red IEEE 802.11. Estos equipos cuentan con una gran potencia de transmisión, por lo que al sumarle la ganancia de la antena empleada, en la mayoría de los casos excede la limitación nacional de la nota UN-85.

Se han encontrado numerosas publicaciones de diferentes investigaciones en las que se utiliza la tecnología Wi-Fi sobre el mar o agua dulce, pero en ellas o no se

---

<sup>8</sup><http://www.wififorboats.com/>

<sup>9</sup><http://www.thewirie.com/>

<sup>10</sup><http://www.ubnt.com/>

<sup>11</sup><http://guifi.net/>

especificaba la distancia de los enlaces, o estos eran de decenas o centenas de metros. Por ello, estos trabajos no se han incluido en el presente documento.

## 2.7. WiMAX

WiMAX son las siglas de Worldwide Interoperability for Microwave Access, y es la marca que certifica que un producto está conforme con los estándares de acceso inalámbrico IEEE 802.16. El único organismo habilitado para realizar dicha certificación y garantizar la interoperabilidad entre equipamiento de distintos fabricantes es el WiMAX Forum. El estándar actual es el IEEE 802.16-2009, que incluye tanto aplicaciones fijas como móviles. WiMAX puede operar en Europa en las bandas sin licencia de 5.470 a 5.725 GHz y de 5.725 a 5.825 GHz, siendo posible emplear mayor potencia en la primera. Esta tecnología puede llegar a alcanzar distancias de hasta 80 kilómetros con antenas muy direccionales y de alta ganancia, y velocidades de hasta 40 Mbps, en condiciones óptimas.



Figura 2.5: Logo WiMAX Forum.

Pese a que en versiones anteriores del estándar se contempló un modo de operación *Mesh Networking*, actualmente este modo ha sido eliminado. Por otro lado, el estándar IEEE 802.16j-2009, también conocido como *multihop relay*, permite que cada nodo actúe como estación repetidora (RS) o como estación subscriptora (SS), pudiéndose extender el alcance de las redes y operar con topología de malla (ver Figura 2.6). Además, WiMAX permite la utilización de las modulaciones OFDM (*Orthogonal Frequency-Division Multiplexing*) y OFDMA (*Orthogonal Frequency-Division Multiple Access*) para una mejor operación en enlaces NLOS.

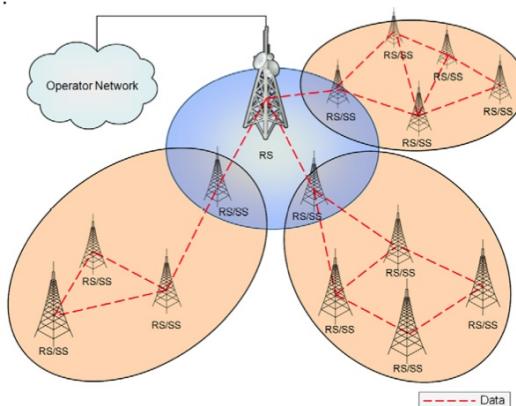


Figura 2.6: Topología Multihop Relay de WiMAX.

Esta tecnología ha tenido una gran acogida en aplicaciones para el sector marítimo ya que una de las más crecientes demandas de la industria naval es la necesidad de disponer de comunicaciones de banda ancha fiables, robustas y económicamente competitivas. Por ello, existen varios grupos de investigación trabajando en este campo. Entre ellos destacamos el proyecto TRITON, cuyo objetivo es crear una red de malla inalámbrica entre los barcos próximos a la costa para dar acceso de banda ancha a Internet (ver Figura 2.7a). En ella cada barco debe comportarse como un nodo capaz de retransmitir la información que le llega a hacia un determinado destino. Este grupo cuenta con un gran número de publicaciones que abarcan gran cantidad de aspectos importantes como estudios de: pérdidas de propagación en puertos marítimos, protocolos de acceso al medio, interferencias entre nodos vecinos, protocolos de encaminamiento, etc. (Pathmasuntharam et al., 2008) (Kong et al., 2009a) (Kong et al., 2009b) (Zhou et al., 2010)

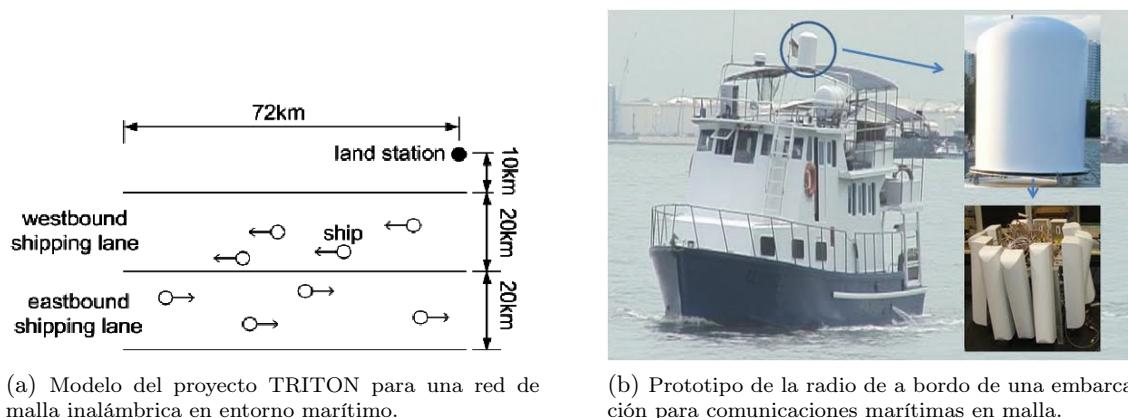


Figura 2.7: Imágenes tomadas del proyecto TRITON. (NICT©)

Se destaca de este trabajo que la mayoría de las publicaciones están basadas en simulaciones y modelos de la realidad, y apenas proporcionan algún resultado de medidas de campo. En la Figura 2.7b, se muestra una imagen extraída de la página web del Dr. Ming-Tuo Zhou, investigador de TRITON, en la que aparece un prototipo de la radio de a bordo de una embarcación para comunicaciones marítimas en malla. Este consiste en un array de 12 antenas direccionales, lo que logrará un gran alcance en todas las direcciones.

La empresa española Quobis, ha realizado alguna medida empírica del alcance de esta tecnología sobre el medio marino (Pérez et al., 2006). Concretamente, en la banda de 5.4 GHz y empleando antenas direccionales de ganancia 21 y 28 dBi.

Inicialmente realizan dos enlaces, de 5 y 11 kilómetros, con la práctica totalidad del vaneo del enlace discurriendo sobre la superficie del mar, pero con las estaciones estáticas en la costa. En ambos casos consiguen una conexión estable con un throughput neto cercano a los 7 Mbps, valor nominal anunciado por el fabricante para los equipos que utilizan. En una de las pruebas, observaron que variaciones en el apuntamiento o acimut de una de las antenas provocaban disminuciones en la tasa neta de transferencia de unos 2 Mbps, pero sin llegar a perder la conexión. Este efecto probablemente se deba a que las antenas de alta ganancia tiene un ancho de haz pequeño, y por lo tanto son muy sensibles al desapuntamiento.

Finalmente realizan una última prueba situando una de las antenas en una em-

barcación a unos 3 metros sobre la línea de flotación. Dicha embarcación realizan un recorrido circular, llegando a alejarse hasta unos 8 kilómetros de la estación en la costa. Durante el recorrido se observan rápidas fluctuaciones sobre el valor medio registrado de throughput, unos 6,5 Mbps. Estas variaciones, que llegar a ser de hasta 5 Mbps, son debidas al efecto del oleaje en la embarcación, que hace variar la dirección de apuntamiento de la antena.

Garropo et al. (2009) presentan los resultados de una campaña de medidas llevadas a cabo en entorno marítimo con tecnología WiMAX operando en 2.5 GHz. Posteriormente, comparan los resultados obtenidos con el modelo teórico de propagación a dos rayos, demostrando una gran aproximación entre ambos.

Para las pruebas sitúan la estación base en la costa en el tejado de un edificio a unos 30 metros. Utilizan una antena sectorial de ganancia 17 dBi, alcanzando finalmente una PIRE (*Potencia Isotrópica Radiada Equivalente*) de 52 dBm.

El otro equipo se encuentra en una embarcación, con la antena situada a unos 10 metros sobre la línea de flotación. En este caso utilizan una antena omnidireccional de 11 dBi de ganancia. El procedimiento consistió en transmitir desde la costa al barco mientras este se iba alejando en línea recta. Observaron caídas periódicas de la potencia recibida de mas de 20 dB hasta aproximadamente los 5 kilómetros, donde comienza a darse un comportamiento estable con unas pérdidas de 1 dB/Km. El punto donde deja de haber cancelaciones periódicas de la señal recibida se conoce como punto de ruptura o *breakpoint*. La señal recibida tuvo una potencia superior a los -60 dBm hasta los 14 kilómetros. La máxima distancia de conexión que lograron fue 19.2 Km.

Aunque en los citados casos se habla de enlaces reales de distancias muy apropiadas para el objetivo de este proyecto, se destaca que en la mayoría de los casos se han empleado antenas muy direccionales, a alturas por encima de los 2 metros sobre la línea de flotación y, en algunos casos incluso, empleando una PIRE por encima de lo permitido en España. Estas características no podrán replicarse en el caso de implementarse sobre boyas, por lo que no se puede esperar obtener los mismos resultados en caso de decantarse por la utilización de sistemas WiMAX.

## 2.8. ZigBee

ZigBee es el nombre, creado por la ZigBee Alliance, de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización en radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal, que únicamente define la capa física y MAC (ver Figura 2.8).

Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

ZigBee utiliza las bandas ICM, en concreto, 868 MHz en Europa, 915 MHz en Estados Unidos y 2.4 GHz en todo el mundo, con tasas de 20, 40 y 250 kbps respectivamente. Se definen tres tipos distintos de dispositivos: coordinadores, routers y dispositivos finales, y atendiendo a su funcionalidad se clasifican en dos grupos: FFD (*Full Function Device*), al que pertenecen los coordinadores y routers, y RFD (*Reduced Function Device*), que son los dispositivos finales (ver Figura 2.2a).

Uno de los campos en los que mayor aceptación ha tenido esta tecnología han sido en las redes de sensores inalámbricas (WSN), gracias a:

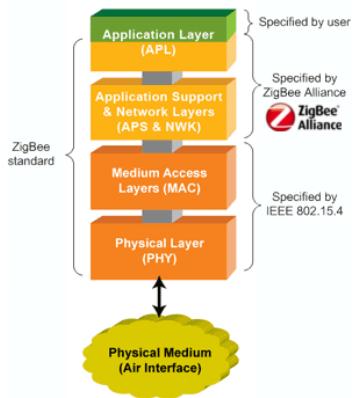


Figura 2.8: Pila de protocolos ZigBee.

- Su topología de red de malla. Una red ZigBee puede constar de un máximo de 65535 nodos distribuidos en subredes de 255 nodos.
- Su bajo consumo. Los nodos soportan un modo *sleep* en el que el consumo es muy bajo y, generalmente, se encuentran en dicho modo la mayor parte del tiempo.
- Su fácil integración, se pueden fabricar nodos con muy poca electrónica.

Aunque la potencia de salida típica de los módulos ZigBee suele ser de 0 dBm, logrando alcances que oscilan entre los 10 y 75 metros, dependiendo bastante del entorno, actualmente también se han desarrollado módulos ZigBee de largo alcance que emplean una potencia mayor logrando enlaces incluso de kilómetros.

Existe un grupo de investigación que ha trabajado en la implementación de una red de sensores en el mar para monitorizar los niveles de temperatura (Sieber et al., 2008) (Sieber et al., 2009). Para ello, emplean boyas que permiten colocar las antenas a una altura aproximada de 1.5 metros sobre la línea de flotación. En las publicaciones detallan pruebas de alcance realizadas con dos tipos de módulos de comunicación distintos.

En primer lugar, los módulos emplean tecnología ZigBee y operan en la banda de 2.4 GHz. Transmiten con una PIRE de 20 dBm y tienen una sensibilidad de -104 dBm. La máxima distancia de comunicación que lograron en este caso fue de 1 kilómetro, a pesar de que el fabricante hablaba de hasta 4 Km.

En segundo lugar, utilizan módulos RF que operan en la banda de 868 MHz. Estos módulos no son ZigBee, tienen una potencia máxima de salida de 25 dBm y una sensibilidad de -112 dBm. Con ellos logran implementar un enlace de 3 kilómetros, pero hablan de que no pudieron continuar con la prueba y que en un futuro tratarían de probarlo a mayor distancia.

Zennaro et al. (2010) realizan una campaña extensa de medidas en la que utilizan siete tipos de módulos de comunicación distintos. Estas pruebas se llevan a cabo para seis distancias diferentes y condiciones LOS y NLOS. El lugar de las pruebas es el desierto de Los Monegros (Zaragoza). Los resultados obtenidos se recogen en la Figura 2.9, que muestra los siguientes datos: protocolo, frecuencia de operación, potencia de transmisión y sensibilidad de cada uno de los módulos, ganancia de las antenas utilizadas (2 y 5 dBi), distancia y condiciones de visión de cada enlace, y el throughput y RSSI (*Received Signal Strength Indication*, Indicador de Potencia de Señal Recibida)

medido en cada caso. En cada prueba envían 100 paquetes de 90 bytes cada uno, siendo el throughput el porcentaje recibido de ellos.

XBee features	Feature	Dev1	Dev2	Dev3	Dev4	Dev5	Dev6	Dev7
	Protocol	802.15.4	802.15.4	Zigbee-Pro	ZigBee-Pro	RF	RF	RF
	Frequency (Hz)	2.4G	2.4G	2.4G	2.4G	868M	900M	900M
	TX power (mW)	1	63	2	50	315	50	100
	Sensitivity(-dBm)	92	100	96	102	112	100	106
Throughput	Distance	Dev1	Dev2	Dev3	Dev4	Dev5	Dev6	Dev7
2dBi	356m (LOS)	85%	100%	100%	100%	100%	100%	100%
	639m (LOS)	0%	100%	0%	100%	100%	100%	100%
	6363m (LOS)	0%	18%	0%	25%	100%	0%	80%
	12136m (LOS)	0%	0%	0%	0%	100%	0%	0%
	1239m (NLOS)	0%	0%	0%	0%	100%	0%	100%
	3810m (NLOS)	0%	0%	0%	0%	0%	0%	0%
5dBi	356m (LOS)	100%	100%	100%	100%	100%	100%	100%
	639m (LOS)	19%	100%	100%	100%	100%	100%	100%
	6363m (LOS)	0%	100%	0%	100%	100%	0%	100%
	12136m (LOS)	0%	0%	0%	0%	100%	0%	100%
	1239m (NLOS)	0%	0%	0%	0%	100%	0%	100%
	3810m (NLOS)	0%	0%	0%	0%	50%	0%	10%
RSSI(dBm)	Distance	Dev1	Dev2	Dev3	Dev4	Dev5	Dev6	Dev7
2dBi	356m (LOS)	-94	-72	-84	-70	-70	-70	-70
	639m (LOS)		-91		-78	-70	-70	-70
	6363m (LOS)				-97			-94
	12136m (LOS)				-100			
	1239m (NLOS)					-77		
5dBi	356m (LOS)	-87	-70	-72	-70	-70	-70	-70
	639m (LOS)	-94	-70	-90	-70	-70	-70	-70
	6363m (LOS)				-80			-101
	12136m (LOS)		-97		-97			-83
	1239m (NLOS)				-78			-93
	3810m (NLOS)							

Figura 2.9: Resultados obtenidos en Zennaro et al., 2010.

Se puede observar que los resultados para condiciones NLOS son muy negativos para 2.4 GHz, aun utilizando la máxima potencia. No ocurre lo mismo con los módulos de RF en 868 MHz y 900 MHz, que utilizando su máxima potencia logran conexión incluso en todos los casos, aunque menor throughput en condiciones NLOS. Por lo tanto, se concluye que existe un compromiso entre velocidad de envío de datos (2.4 GHz), y alcance y robustez del enlace (868 MHz y 900 MHz).

## 2.9. Otras tecnologías

Fitz et al. (2005) realizan medidas entre dos boyas para las bandas de 173.25 MHz y 433 MHz. Concretamente, miden la potencia de señal recibida en una de ellas a medida que van incrementando la distancia del enlace. Las boyas que utilizan son muy pequeñas, y únicamente permiten colocar las antenas a una altura de 0.1 y 0.2 metros sobre la línea de flotación para 173.25 MHz y 433 MHz respectivamente. En ambos casos las antenas utilizadas son un monopolio  $\lambda/4$ , y transmiten con una PIRE de 0 dBm. Para ambas frecuencias miden una baja tasa de errores de bit hasta pérdidas de 115 dB, valor a partir del cual los errores aumentan rápidamente hasta que se pierde la conexión. Este hecho, lo registran a poco más de 1 kilómetro para 433 MHz, y a 3 kilómetros para 173.25 MHz. A continuación, utilizando varias boyas operando en esta última frecuencia, realizan pruebas de una red multisalto. Los resultados obtenidos son:

- Las boyas que se encuentran a menos de 350 metros del nodo colector consiguen entregar un 95 % de sus paquetes.
- Las boyas situadas a una distancia de 750 metros del nodo colector entregan una media del 55 % de los paquetes, con variaciones entre el 40 % en el peor de los

casos y 75 % en el mejor.

Por último, utilizando los datos medidos en el enlace simple, una estimación de las pérdidas que puede introducir al enlace cada nodo, y los resultados en las pruebas de la red, concluyen que el movimiento de las boyas, junto con otros efectos como la dispersión producida por la superficie del mar, la difracción de la onda o la inclinación de las antenas, llegan a introducir unas pérdidas en el enlace de hasta 25 dB. Se destaca, que la banda de 173.25 MHz empleada en estas pruebas, está restringida en España a servicios fijo y móvil, según la nota UN-98.

Palazzi et al. (2005) realizan un enlace punto a punto de 78 kilómetros sobre la superficie del mar entre dos estaciones fijas. La frecuencia de operación es 10.5 GHz, las antenas se encuentran a una altura aproximada de 5 metros, la PIRE de la estación transmisora es 30 dBm y la estación receptora cuenta con una antena de 30 dB de ganancia y un amplificador de ganancia de 50 dB conectado finalmente a un analizador de espectro. Tal como predicen inicialmente tras un estudio teórico, en recepción logran una SNR (*Signal to Noise Ratio*, Relación Señal-Ruido) de 40 dB. En la publicación, comentan que superan en 10 dB la PIRE máxima permitida en Australia para dicha banda. En España, según las notas UN-86 y UN-137, la banda de 10.5 GHz se considera de uso común, pero no para su aplicación en comunicaciones.

Existen algunos proyectos que utilizan boyas de gran tamaño en el mar para diversas aplicaciones y que emplean comunicaciones por satélite para hacer llegar sus datos a la costa. Por ejemplo las boyas desarrolladas por el *Woods Hole Oceanographic Institution (WHOI)*<sup>12</sup>. Por su elevado coste y la necesidad de depender de sistemas ajenos, las comunicaciones por satélite no se contemplan por el LAB como opción para sus sistemas de PAM.

---

<sup>12</sup><http://www.whoi.edu/>

# Capítulo 3

## Definición del prototipo de red ad hoc

*Un viaje de mil leguas comienza con un paso.*

Tao Te King  
Capítulo LXIV  
Lao Tse

**RESUMEN:** Este capítulo comienza presentando los factores determinantes y las características deseables para la implementación de la red ad hoc de boyas en el mar. A continuación, se analizan las posibles tecnologías de comunicación inalámbrica para su utilización en el sistema y se selecciona justificadamente una de ellas. Por último, se muestra la arquitectura de una boya definida por el LAB y se presentan los nuevos módulos definidos en este proyecto para permitir su operación en red, sentando las bases para tratar temas más técnicos en los próximos capítulos.

### 3.1. Factores determinantes

Se ha tratado de concretar una serie de factores que determinan diferentes aspectos del proyecto. A continuación, se presenta cada uno de ellos:

- **Tipo de boyas.** Este factor será clave para la definición global del sistema. Se diferencian principalmente dos grupos: las boyas de vara o similares (Figura 3.1a), y las estructuras flotantes de gran dimensión (Figura 3.1b). Este segundo grupo supone un elevado coste de adquisición o desarrollo, así como de despliegue y recogida mediante grandes barcos. Debido a que determinadas aplicaciones, como la instalación de perímetros de seguridad en campañas de prospección petrolífera, requieren la movilidad de las boyas con periodos de algunas horas; y que el LAB persigue desarrollar una herramienta de mitigación viable y sostenible en todos sus aspectos, incluido el económico, inicialmente se trabajará con el primer tipo de boyas. El tamaño y la forma concreta de éstas dependerá del número de baterías, del tipo de paneles fotovoltaicos que se utilicen finalmente y del peso total que tendrá que mantenerse a flote.

El tipo de boyas determinará aspectos como:



(a) Boyas de pequeña dimensión.



(b) Boyas de gran dimensión.

Figura 3.1: Tipos de boyas según el tamaño.

- La altura máxima sobre la superficie del mar a la que podrá situarse la antena, y la movilidad que ésta tendrá respecto a su posición ideal, que se considera perpendicular a la superficie. Como se vió en la Sección 2.5, interesa poder situar las antenas en las boyas de la red a la mayor altura posible sobre la línea de flotación. De esta manera se logra un mayor despejamiento de la primera zona de Fresnel y, por lo tanto, la contribución destructiva de los rayos reflejados será menor. Empleando el tipo de boyas comentado en el punto anterior, se estima la posibilidad de colocación de las antenas a una altura aproximada de 2 a 3 metros sobre la línea de flotación, y éstas sufrirán un movimiento considerable, que aumentará con la altura a la que se encuentren.
- La cantidad de energía que podrá generarse y almacenarse. Las baterías tienen un peso elevado y el número de ellas que pueda contener una boya dependerá del peso que ésta pueda soportar y del volumen de que disponga para albergarlas. Además, el número y tamaño de paneles fotovoltaicos también se verá determinado por el tipo de boya y, por lo tanto, la energía generada estará limitada.
- **Energía disponible.** Como se ha visto dependerá del tipo de boya utilizada pero, a su vez, la energía disponible será un factor determinante de:
  - La capacidad de procesamiento del audio capturado por los hidrófonos que tendrá cada boya para detectar los eventos acústicos.
  - La autonomía, o tiempo que podrá estar operativa sin necesidad de supervisión técnica o mantenimiento.
  - El sistema de comunicación a utilizar, ya que no todos tienen el mismo consumo energético.
- **Frecuencia de operación.** Como se explicó en el Capítulo 2, la banda de operación será sin licencia. La elección de una u otra determinará aspectos importantes como:
  - El tipo de servicio que soporte la herramienta. Como se vió en la Sección 2.8, existe un compromiso entre la velocidad de envío de datos, siendo esta mayor a frecuencias altas, y el alcance y robustez del enlace, que es mayor a

frecuencias bajas. Por ello, si un determinado servicio requiere el envío de una cantidad importante de datos será necesario utilizar frecuencias más elevadas y, como consecuencia, las boyas deberán estar más próximas entre sí. En caso de enviar pocos datos se podrá operar con frecuencias más bajas, que permitirán aumentar la distancia entre boyas.

- Los módulos de comunicación a emplear. Según la tecnología, éstos tendrán un coste mayor o menor. Lo mismo ocurre con el consumo energético. Si este es elevado, conlleva varias limitaciones que, en caso de poder superarse, implican una mayor inversión.
- La potencia máxima de transmisión y otros factores que, como se vió en la Sección 2.1, vendrán impuestos por las limitaciones de las correspondientes administraciones.

### 3.2. Características deseables

A la hora de evaluar y elegir la tecnología más apropiada para la consecución del proyecto, se ha considerado que ésta debe de reunir como características más deseables las siguientes:

- **Bajo costo.** La herramienta final debe ser lo más económica posible, entendiendo por ello el coste que supone monitorizar una determinada área, y no el coste unitario de cada boya. Esto se recalca ya que, casi siempre, al optar por algún componente concreto, condiciona la elección de otros. Por ejemplo, si el sistema de comunicación consume mucha potencia serán necesarios paneles fotovoltaicos mejores y baterías con mayor capacidad, por lo tanto, aunque el primero sea económico, el resultado final puede verse encarecido. La viabilidad económica de la herramienta es un factor muy importante, ya que determinará el que ésta sea empleada por las empresas para mitigar los efectos del ruido antropogénico en el mar, derivado de su operación, o sea rechazada por un coste muy elevado.
- **Bajo consumo.** Al tratarse de boyas autónomas que operarán de manera aislada en el mar, se debe tratar de reducir el consumo de energía todo lo posible ya que, como se comentaba antes, un gran consumo supondrá la necesidad de un sistema mejor para proporcionar dicha energía, lo que a su vez puede requerir la utilización de una boya de mayor tamaño y, por consiguiente, a mayor consumo mayor será el coste total del sistema.
- **Gran alcance.** El número de boyas necesarias para monitorizar acústicamente una determinada zona dependerá inversamente del alcance o área cubierta por cada una de ellas. Si éste es pequeño será necesario colocar las boyas más próximas entre sí, sobremuestreando innecesariamente el medio e introduciendo más tráfico a la red. Por ello, en caso de que el alcance sea menor de lo deseado, se plantea la utilización de boyas repetidoras, que no capturan ni procesan sonido y únicamente actúan de nexo entre otras que sí lo hacen. En definitiva, dependiendo del número de boyas necesarias, habrá que estudiar en cada caso qué opción es más apropiada.
- **Soporte de topología ad hoc.** El hecho de que todos los nodos deban tener conexión directa con un nodo central o con una estación en tierra limita mucho la extensión de la red. Por ello, se quiere implementar una red ad hoc, de manera

que cada nodo pueda enrutar su propia información y la de sus nodos vecinos hacia un destino concreto, buscando en cada caso el camino más adecuado. Esto dota al sistema de mayor robustez, ya que si algún nodo se cae, el resto de la red es capaz de continuar operando. Esta característica es muy importante en un medio aislado, como es el mar. Se tendrá en cuenta como factor importante que una tecnología ya tenga implementada dicha opción y únicamente requiera activarla y configurarla.

- **Naturaleza del software.** Se valorará positivamente que la opción resultante sea open source. A parte de un menor coste, esto permitirá corregir, mejorar o modificar cualquier aspecto que se estime oportuno.
- **Alta tasa de transferencia de datos.** Esta característica es deseable en general pero, según el tipo de servicio final que pretenda darse con la herramienta, será más o menos crítico. Se recuerda que, las redes ad hoc introducen una cantidad importante de información para su operación, lo que hace que el throughput disminuya.

### 3.3. Elección de una tecnología

En la Tabla 2.1 se recogen las actuales bandas ICM definidas por la UIT. Como se indicó anteriormente, el presente proyecto operará en una de estas bandas, o sus equivalentes en España (ver Tabla 2.2), teniendo en cuenta diversos aspectos ya mencionados.

En primer lugar, se descartan las bandas que se encuentran por debajo de 433 MHz. En estas frecuencias tan bajas, la velocidad de envío de datos es muy reducida, y el tamaño de las antenas excesivo para su colocación en una boyas. Por ejemplo, para una frecuencia de operación de 40.70 MHz, una antena cuarto de onda medirá  $\lambda/4$ , es decir, 1.84 metros.

En segundo lugar, también se descartan las bandas por encima de 5.6 GHz. Las razones son: la poca presencia en el mercado de equipos de comunicación que operen en ellas, sus precios elevados y la reducida potencia de transmisión permitida, teniendo en cuenta que las frecuencias altas se ven más afectadas por fenómenos como: la difracción, la refracción, los agentes meteorológicos, etc. (McLarnon, 2001)

Por lo tanto, únicamente se consideran apropiadas las bandas en: 433 MHz, 868 MHz, 2.4 GHz y 5.6 GHz. En la Tabla 3.1 se presenta una comparación de las tecnologías que operan en ellas. El objetivo es poder evaluar su idoneidad para este proyecto de acuerdo con lo expuesto en la Sección 3.2. Los valores recogidos en la tabla son aproximados, ya que varian según los equipos de cada fabricante, e incluso entre distintos modelos de un mismo equipo. Entre los productos consultados para su elaboración se destacan los de las siguientes empresas: *RFSolutions, HopeRF, Digi International, Laird Technologies, Ubiquiti Networks, Alvarion, Albentia Systems, Sloka Telecom, Tranzeo*, entre otras.

En este punto del proyecto se realizó una reunión con el personal del LAB involucrado en el desarrollo de las boyas. El objetivo era concretar las funcionalidades de la herramienta. En este encuentro se acordó lo siguiente:

- Trabajar inicialmente en el servicio de alarmas; es decir, sólo será necesario enviar alarmas a través de la red, y se dejará para más adelante el envío de audio, espectrogramas y otros datos.

- La red de boyas debe comportarse como una red ad hoc.
- La distancia entre boyas debe ser de al menos algunos kilómetros para que la herramienta interese.

Estas conclusiones rompen, en cierta medida, el compromiso existente entre la velocidad de envío de datos y el alcance de la señal. Al trabajar únicamente con alarmas el tráfico de la red será menor y las boyas no requerirán una tasa de envío de datos tan elevada. Aun así, se recuerda que las redes ad hoc introducen un tráfico extra a la red a modo de: mensajes de conectividad, cabeceras, etc.

Tecnología Banda	RF 433 MHz	RF 868 MHz	ZigBee 2.4 GHz	Wi-Fi 2.4 GHz	Wi-Fi 5.6 GHz	WiMAX 5.6 GHz
Estándar	-	-	802.15.4	802.11b/g/n	802.11a	802.16
PIRE máxima en la banda	12.15 dBm	29.14 dBm	20 dBm	20 dBm	30 dBm con TPC 27 dBm sin TPC	30 dBm con TPC 27 dBm sin TPC
Sensibilidad	-100 dBm	-112 dBm	-100 dBm (1% PER <sup>1</sup> )	-97 dBm @ 1 Mbps (< 8 % PER) -72 dBm @ 65 Mbps (< 10 % PER)	-94 dBm @ 6 Mbps -74 dBm @ 54 Mbps	-92 dBm (BPSK-1/2) (BW 10 MHz) -74 dBm (64QAM-3/4) (BW 10 MHz)
Link budget	112.15 dB	141.14 dB	120 dB	117 dB @ 1 Mbps 92 dB @ 65 Mbps	124 dB @ 6 Mbps 104 dB @ 54 Mbps	122 dB (BPSK-1/2) 104 dB (64QAM-3/4)
Tasa transmisión radio	100 kbps	24 kbps (CT <sup>2</sup> 10 %)	250 kbps	1/2/6/9/12/18/24/ 36/48/54/65 Mbps	6/9/12/18/24/ 36/48/54 Mbps	10/14/28/37/73/ 130/250 Mbps
Voltaje de operación (VDC)	2.2 - 3.8	3.0 - 3.6	2.8 - 3.4	3.1 - 3.6	24 PoE <sup>3</sup>	12/24/48 PoE
Consumo máximo de potencia	0.10 W	1.65 W	1.12 W	0.89 W	4 W	15 - 25 W
$I_{TX}$	30 mA	500 mA @ 3.3 V	340 mA	260 mA	-	-
$I_{RX}$	13 mA	65 mA	55 mA	140 mA	-	-
$I_{Sleep}$	1 $\mu$ A	55 $\mu$ A @ 3.3 V	< 50 $\mu$ A	< 2 $\mu$ A @ 25°C	-	-
Peso	-	-	-	-	0.18 kg	2 - 3 kg
Precio (€)	10 - 18	40 - 60	40 - 50	40 - 70	60 - 70	1000 - 4000

Tabla 3.1: Tabla comparativa de las tecnologías disponibles en las bandas: 433 MHz, 868 MHz, 2.4 GHz y 5.6 GHz.

Volviendo a la Tabla 3.1, se pueden hacer las siguientes observaciones:

- Los módulos en 433 MHz y en 868 MHz no siguen ningún estándar, por lo tanto, la compatibilidad entre distintos fabricantes no está garantizada.
- En cada caso se cuenta con operar con la PIRE máxima permitida en la banda. Esta se alcanzará entre la potencia de transmisión del equipo y la ganancia de la antena seleccionada.
- Las tecnologías que pueden operar con varias velocidades son más sensibles cuanto más baja sea ésta.
- El consumo de las tecnologías que operan en 5.6 GHz es considerablemente más elevado que el resto.
- Las tecnologías que no indican sus pesos se corresponden con módulos de tamaño algo mayor que una moneda, por lo que se entiende que éste es muy bajo.
- La tecnología WiMAX, por su peso, consumo energético y precio, se encuentra en otra escala en comparación con el resto. Además, estos mismos motivos, no la hacen muy idónea para su utilización en boyas autónomas.

<sup>1</sup>Packet Error Rate, tasa de error de paquetes recibidos.

<sup>2</sup>Ciclo de trabajo es la fracción de tiempo donde la señal es positiva o se encuentra en estado activo.

<sup>3</sup>Power over Ethernet, alimentación a través de Ethernet.

Finalmente, se decide trabajar en la banda de 868 MHz por los siguientes motivos:

- Al no ser tan crítica la velocidad de envío de datos, lo ideal es optar por frecuencias más bajas, que serán más robustas en este medio y en estas condiciones de movilidad.
- El link budget en esta banda es considerablemente mayor al resto, lo que se traduce en que se soportan más pérdidas entre el emisor y el receptor y, por lo tanto, la distancia entre ellos podrá ser mayor.
- La tasa de envío de datos es suficiente para soportar un protocolo de encaminamiento ad hoc y el envío de alarmas, que no será muy elevado.
- Aunque la banda 868 MHz únicamente es válida en la Región 1 definida por la UIT, en las otras dos regiones existe una banda de características muy similares en 900 MHz. Por ello, se podrán obtener resultados muy similares operando en cualquier parte del mundo.
- El consumo energético, precio y peso de esta tecnología son bajos y, por ello, ideales para su utilización en boyas autónomas.
- Como se citó en la Sección 2.8, existen casos reales en los que se ha probado esta tecnología en condiciones muy similares a las que persigue este proyecto y con resultados positivos. (Sieber et al., 2008) (Sieber et al., 2009)

Aun así, no se descartan el resto de posibilidades y, en un futuro, se espera poder realizar pruebas con otras tecnologías para comparar resultados y, finalmente, lograr una herramienta óptima.

### 3.4. Visión global del sistema LIDO DCL implementado sobre boyas autónomas

Actualmente, la arquitectura que el LAB ha definido para implementar su sistema LIDO DCL sobre boyas autónomas se muestra en la Figura 3.2.

Esta arquitectura, que recuerda mucho a la de los nodos de las redes de sensores inalámbricas, está compuesta por los siguientes subsistemas:

- **Subsistema de energía.** Integrado por un módulo de generación de energía mediante paneles fotovoltaicos y otro de almacenamiento en baterías. Este subsistema se dimensionará en función del consumo energético total que suponga la operación de la boyá, y del tiempo de autonomía que interese.
- **Subsistema sensorial.** La parte sensora de la boyá es el hidrófono. Éste captura el sonido subacuático y lo convierte en una señal eléctrica, que es la entrada del subsistema computacional. Se contempla la posibilidad de utilizar varios hidrófonos a distintas profundidades para, entre otras cosas, facilitar la localización de las fuentes.
- **Subsistema radio.** Formado por todos los sistemas radioeléctricos que utilice la boyá. Inicialmente cuenta con un sistema GPS que permite conocer las coordenadas geográficas de la boyá, ya que ésta podrá estar sin anclaje. Esta información será útil para el servicio de alarmas, pudiendo conocer la posición de la boyá que la genera, y también para la recogida de estas al finalizar su utilización.

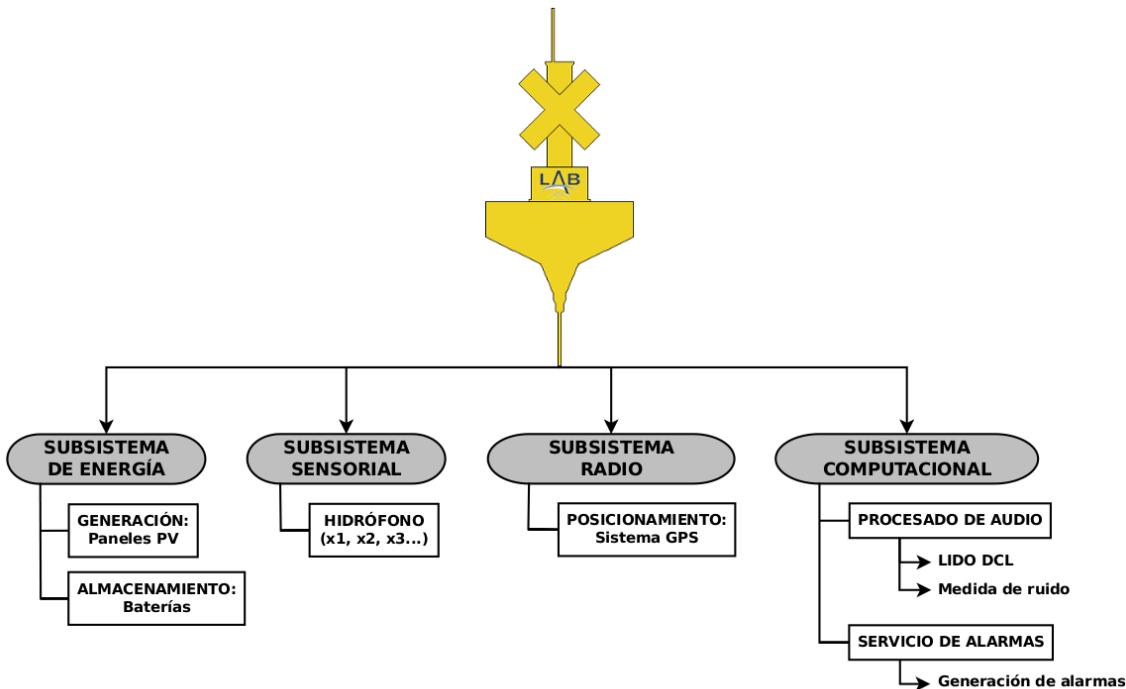


Figura 3.2: Arquitectura de la implementación del sistema LIDO DCL sobre una boyta autónoma.

- **Subsistema computacional.** Cada boyta contará con un *PC embebido* con el sistema operativo GNU/Linux. Este ordenador tendrá varios módulos o programas en ejecución.

En primer lugar, se realizará el procesado del audio capturado por el hidrófono. Para ello se emplea el software LIDO DCL ya desarrollado, encargado de la detección, clasificación y localización de fuentes acústicas, y también un módulo de medida de ruido.

En segundo lugar, estará en ejecución el servicio de alarmas, que en función de sus parámetros de entrada generará alarmas al detectar ciertos eventos acústicos. El objetivo de estas alarmas será alertar de la presencia de cetáceos en una determinada zona.

### 3.5. Arquitectura del sistema LIDO DCL sobre una boyta autónoma para operar dentro de una red ad hoc

En base a lo expuesto en las secciones anteriores, a continuación, se define el prototipo de red a implementar, incluyendo la tecnología que se utilizará y cómo se integrará con el sistema LIDO DCL ya desarrollado. En la Figura 3.3 se muestra la arquitectura que tendrá cada boyta autónoma que actúe como nodo de la red ad hoc. En recuadros verdes con el borde discontinuo se muestran los nuevos módulos definidos en este proyecto, y que se integran con la arquitectura ya definida en la Sección 3.4.

Dentro del subsistema radio, se incorpora un módulo de comunicación, formado por un transceptor que opera en la banda de 868 MHz. Este módulo será el interfaz que conectará la boyta a la red. En el Capítulo 4 se tratará en detalle este tema.

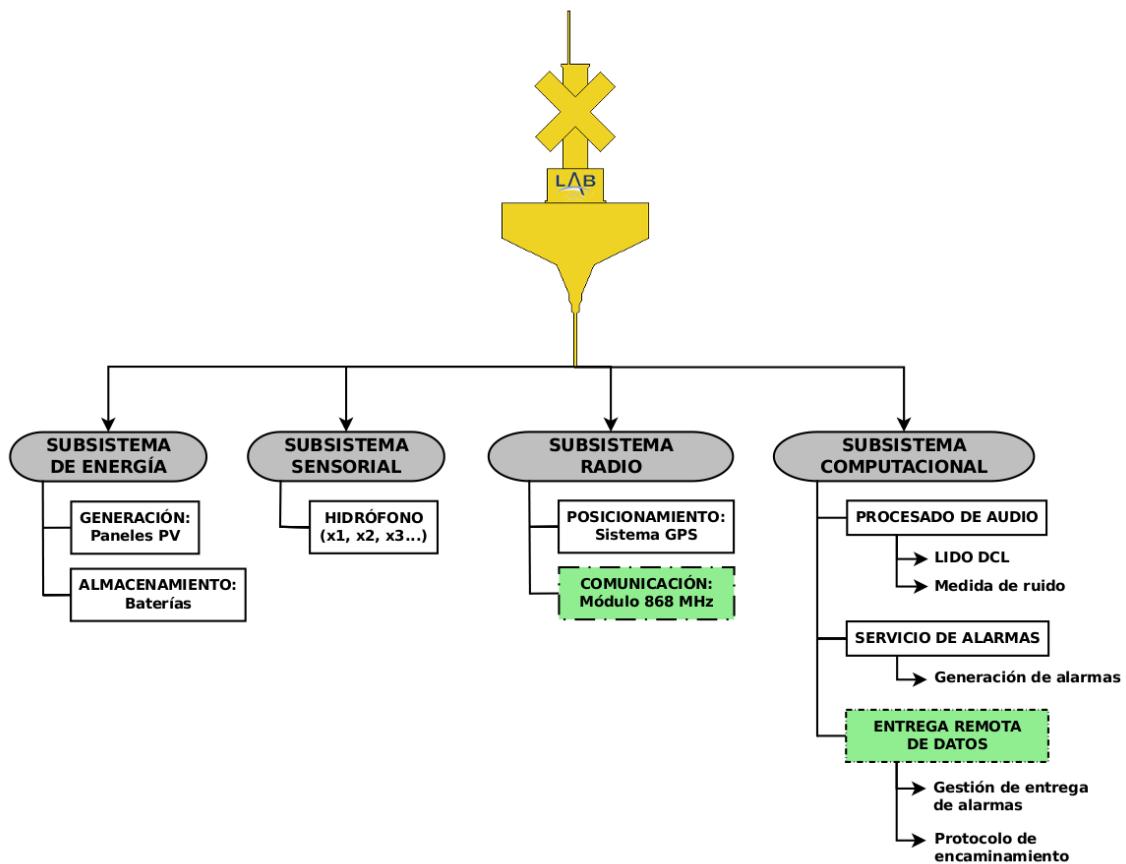


Figura 3.3: Arquitectura del sistema LIDO DCL sobre una boyá autónoma para operar dentro de una red ad hoc.

Por otro lado, dentro del subsistema computacional, se incluye un módulo de entrega remota de datos. Éste se encargará de la gestión de entrega de alarmas generadas por el servicio de alarmas, es decir, de asegurar que éstas lleguen a su destino. Este módulo podrá requerir la utilización de un protocolo de encaminamiento para descubrir rutas hacia el colector, ya sea para enviar alarmas propias o para reenviar las de nodos vecinos. La necesidad de esta última parte dependerá de si ya viene, o no, implementada en el transceptor de 868 MHz.

Con estas novedades, se posibilitará la operación de las boyas sensoras en una red ad hoc, dando lugar a una herramienta de mitigación de los efectos de la actividad humana en el mar, y la problemática que esta supone, como se describió en la introducción del proyecto. En los próximos capítulos, se profundizará en la parte hardware y software de estos nuevos módulos y se presentarán medidas de alcance entre nodos, simulaciones de la red y pruebas reales del sistema que permitirán evaluar, sobre una base experimental, la viabilidad y funcionalidad de esta herramienta de mitigación.



# Capítulo 4

## Hardware

*This new form of communication could have some utility.*

Guglielmo Marconi

**RESUMEN:** El capítulo comienza con un análisis de mercado de equipos de comunicación que operan en la banda 868 MHz seleccionando, finalmente, los módulos XBee-PRO 868 de Digi International para la implementación del prototipo de red. A continuación, éstos se describen en profundidad y, tras ser adquiridos, se configuran y prueban utilizando el software gratuito X-CTU del mismo fabricante. Por último, se realiza un estudio teórico de un enlace radioeléctrico en el mar, y los resultados se comparan y relacionan con medidas reales efectuadas con los módulos seleccionados en dicho entorno.

### 4.1. Análisis de mercado y elección del equipo

Una vez decidida la banda de operación, se realizó un estudio de mercado orientado a la elección de los equipos. Aunque una gran parte de los módulos de comunicación que operan en la banda ICM de 868 MHz tienen una potencia máxima de salida de 10 dBm (10 mW), existen algunos que operan por encima de ésta permitiendo, junto con la correspondiente antena, alcanzar el límite de PIRE de 29.15 dBm (822 mW) impuesto en la banda. En la Tabla 4.1 se muestran las características técnicas de cinco módulos de comunicación destacados del estudio de mercado.

A la hora de decantarse por uno u otro equipo, al tratarse este proyecto de la realización de un prototipo que sirva de primera aproximación a la herramienta y permita evaluar su viabilidad, se consideran factores importantes para la selección:

- El precio, ya que no debe suponer una gran inversión.
- El tiempo de desarrollo hardware y software que implicará el poder comenzar a utilizar el equipo. Por ello, se valora positivamente los módulos cuyo firmware ya permite la operación en malla, soportando el multisalto en la red, y aquellos que requieran un menor montaje previo a su empleo.

El módulo RF301PC1/U1 de Synapse utiliza el firmware SNAP, que soporta la topología de malla. Actualmente éste sólo está disponible con antena chip, lo que

Equipos	Fabricante Modelo	ADEUNIS TWINO HP868	SYNAPSE RF301PC1/U1	DIGI INT. XBee-PRO® 868	QUASAR HM-TRP868D	TELIT TinyOnePro868MHz
<b>Rendimiento</b>	Potencia de transmisión	27 dBm	13 dBm	25 dBm	20 dBm	27 dBm @ 3.6 V
	Sensibilidad	-110 dBm	-121 dBm @ 9.6 kbps (10 % BER)	-112 dBm (1 % PER)	-117 dBm @ 1.2 kbps	-105 dBm 4.8 kbps (CER $10^{-3}$ )
	Link Budget	137 dB	134 dB	137 dB	137 dB	132 dB
	Alcance	6 km	500m / 1.6km	40 km	1 km	4 km
	Tasa transmisión	10/38.4/ 57.6 kbps	150 kbps	24 kbps	1.2 - 115.2 kbps	38.4 kbps
<b>Interfaz serie</b>	Tipo	UART (x3) $I^2C$ (x2) SPI (x2)	UART	UART	UART RS232	RS232
	Velocidad	-	-	1.2 - 230.4 kbps	1.2 - 115.2 kbps	1.2 - 115.2 kbps
<b>Requisitos de energía</b>	Voltaje (VDC)	3 - 3.6	2.7 -3.6	3.0 - 3.6	2.4 - 3.6	3.0 - 3.6
	$I_{TX}$	600 mA	30 mA	500 mA	120 mA	600 mA
	$I_{RX}$	18 mA	18 mA	65 mA	30 mA	30 mA
	$I_{Sleep}$	20 $\mu$ A	2 $\mu$ A	55 $\mu$ A	2 $\mu$ A	< 5 $\mu$ A
<b>Creación de redes</b>	Topología soportada	P2P, P2MP	P2P, P2MP y malla	P2P, P2MP	P2P, P2MP	P2P, P2MP y malla
	Firmware propietario	-	SNAP	-	-	M-ONE
<b>Otros</b>	Conector de antena	I/O 50 $\Omega$	Antena chip / conector U-FL	Wired whip, U-FL y RPSMA	I/O 50 $\Omega$	I/O 50 $\Omega$
	Modulación	GFSK	GFSK	-	FSK	GFSK
	Disponibilidad	SI	SI / Q3 2011	SI	SI	SI
	Precio €	80	-	57	25	90

Tabla 4.1: Características técnicas de módulos de comunicación de alta potencia en la banda 868MHz.

limita mucho sus posibilidades de alcance. El lanzamiento al mercado del modelo con conector U-FL está previsto para el tercer cuatrimestre del 2011, por lo que este módulo se descarta para el proyecto aunque, en un futuro, con los nuevos modelos, podría ser una buena solución.

El otro módulo que soporta topología de malla, gracias a su firmware M-ONE, es el TinyOne Pro 868MHz de Telit. Este equipo está disponible en dos versiones. La primera opción es un terminal directamente preparado para su conexión y utilización. Éste, supondría una gran solución tanto técnica como en términos de tiempo, ya que únicamente requeriría una configuración previa. El inconveniente por el que se descarta el terminal es su elevado precio, 322€, que elevaría el coste final de la herramienta de mitigación y por lo tanto afectaría a su viabilidad y aceptación en la industria.

La segunda opción es su versión en módulo, que requiere la incorporación de un conector de antena y de un interfaz para su conexión con el ordenador. Otra limitación que presenta este equipo, en comparación con otros de la tabla, es que su potencia de transmisión es de 27 dBm por lo que, únicamente, se podría aumentar en 2 dB con

una antena para no exceder la limitación de la banda. Otros equipos tienen un margen mayor hasta la PIRE máxima, lo que permitirá lograr un link budget varios decibelios mayor que en este caso. Finalmente, su precio es de 90€, superior al de otros módulos. Por todo ello, también se descarta la utilización del TinyOne PRO 868 MHz de Telit, tanto en su versión terminal como en módulo.

Los tres módulos restantes implicarán la utilización de un protocolo de enrutamiento ad hoc que irá implementado en el PC embebido. Este tema se tratará en detalle en el Capítulo 5, junto con el resto de la parte software.

El Twino HP868 de Adeunis, y el HM-TRP868D de Quasar, tienen el mismo inconveniente que la versión módulo del TinyOne Pro 868 MHZ. Ambos requieren la implementación de un interfaz para su conexión con el ordenador y, también, la incorporación de un conector para la antena. Además, el modelo de Adeunis tiene un precio elevado en comparación con los otros y la misma limitación para aumentar su link budget debido a la PIRE máxima de la banda. Por ello, el Twino HP868 también se descarta.

Entre los restantes está el XBee-PRO 868 de Digi International y el HM-TRP868D de Quasar. Se opta por el primero, a pesar de ser de entre los dos el más caro. Los motivos son:

- Se trata de un módulo *out of the box*; es decir, únicamente requiere su conexión mediante un adaptador UART-USB al ordenador, acoplarle la antena y configurarle.
- El fabricante dispone de un software gratuito llamado X-CTU para facilitar su configuración en modo gráfico, en vez de por comandos. Este software, además, dispone de una herramienta para realizar pruebas de alcance entre dos de estos módulos.
- Su firmware permite una gran variedad de opciones, como: asignar direcciones a los módulos, utilizar repeticiones de envío a nivel MAC, enviar acuses de recibo, emplear encriptación de los datos, etc.
- Se trata de un módulo muy popular, existen numerosos proyectos documentados realizados con él y diferentes adaptadores UART-USB disponibles en el mercado compatibles con su patillaje.
- Hay una versión idéntica que opera en 900 MHz, lo que permitiría migrar fácilmente de una a otra banda según la región en la que se utilice.
- Se tiene constancia de su utilización en el medio marino entre boyas, logrando hasta 3 kilómetros de alcance (Sieber et al., 2008) (Sieber et al., 2009). Éste es un hecho importante ya que, como se puede observar en la tabla, a pesar de que todos los módulos tienen un link budget similar y utilizan modulación en frecuencia, los alcances indicados en las hojas técnicas de los fabricantes varían entre 1 kilómetro y 40. Por ello, conocer casos reales de su utilización permite tener una idea más fiable de sus prestaciones.

## 4.2. XBee-PRO® 868 RF Modules

### 4.2.1. Descripción

Los módulos XBee-PRO 868, como se mostró en la sección anterior, poseen una potencia de transmisión y una sensibilidad elevadas, lo que les hace ideales para su utilización en largas distancias. Además, su reducido tamaño, 2.443 x 3.322 cm, y su bajo consumo le convierten en una solución apropiada para su utilización en sistemas aislados.

El módulo cuenta con dos buffers internos independientes, que son:

- Buffer de entrada serie. Almacena la información que llega por el pin de entrada de datos, llamado DIN, hasta que ésta pueda ser procesada.
- Buffer de salida serie. Guarda la información RF recibida y la envía por el pin de salida de datos o DOUT, cuando éste está libre para usarse.

Gracias a estos dos buffer el equipo puede actuar como transceptor, recibiendo y enviando datos sin que estos se pierdan.

Los módulos XBee-PRO 868 disponen de dos modos de operación:

- **Transparente.** En este modo se comportan como una línea serie, de modo que, todos los datos UART recibidos por el pin DIN se encolan para su transmisión. Por otra parte, la información RF recibida se envía hacia el ordenador a través del pin DOUT. Los módulos cuentan con una función de paquetización que se encarga de enpaquetar la información del buffer de entrada serie. Estos paquetes, que tendrán un tamaño máximo de 256 bytes, se envían cuando:
  - No se reciben más caracteres por el pin DIN durante un timeout determinado por el usuario en la configuración.
  - Se alcanza el tamaño máximo de paquete.
- **API (*Application Programming Interface*).** Este modo está basado en la utilización de tramas ya definidas. La información debe llegar al módulo mediante comandos y éste se encargará de crear las tramas. Por ello, resulta algo más complejo de utilizar que el transparente. Este modo permite una utilización más completa de las funciones definidas en el firmware del módulo, y facilita algunas operaciones como:
  - Envío de datos a múltiples destinatarios.
  - Control de paquetes recibidos correcta o incorrectamente.
  - Identificación del remitente de cada paquete.

Debido a que el firmware del módulo no soporta la formación de redes multisalto, éstos se utilizarán en modo transparente y enviarán todos sus mensajes en difusión o broadcast. Como ya se comentó anteriormente, la capa de red o el protocolo de encaminamiento se implementará en el PC embebido, por lo que los mensajes llegarán al XBee con sus correspondientes cabeceras y listos para ser enviados.

En la Figura 4.1, se puede observar una vista superior y otra inferior del módulo. En la segunda, una pegatina indica cierta información relevante como: el part number o referencia del fabricante, la revisión del firmware, el número de serie y la fecha y

hora en la que se fijó esa información. En concreto, los módulos adquiridos tienen la última revisión del firmware, designada con la letra G. Ésta se realizó el 22 de Febrero de 2011.

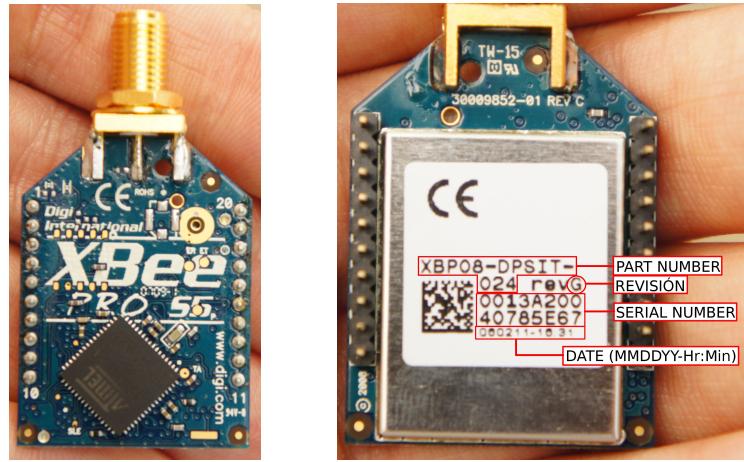


Figura 4.1: Vistas del módulo XBee-PRO 868.

Para la conexión del XBee-PRO 868 al ordenador se ha optado por utilizar un conversor UART-USB (ver Figura 4.2a). Éste está especialmente diseñado para operar con todas las unidades XBee de Digi International. Concretamente, utiliza el integrado FT232RL, conocido como solución para reemplazar el puerto RS232 por el USB en nuevos diseños. Viene con drivers libres de royalties para la mayoría de los sistemas operativos, entre ellos Windows y GNU/Linux. Además, desde Agosto de 2010, estos conversores incluyen el regulador de potencia MIC5219, que convierte los 5V del USB en 3.3V para la alimentación del XBee, y soporta corrientes de hasta 500 mA, necesarias en los modelos PRO.

También se han adquirido antenas de 4.5 dBi de ganancia con conector RP-SMA, compatible con el XBee-PRO 868. De esta manera, se alcanza una PIRE de 29.5 dBm, ajustada al límite permitido en la banda. En la Figura 4.2b se puede observar un nodo formado por el módulo de comunicación, la antena y el conversor UART-USB.

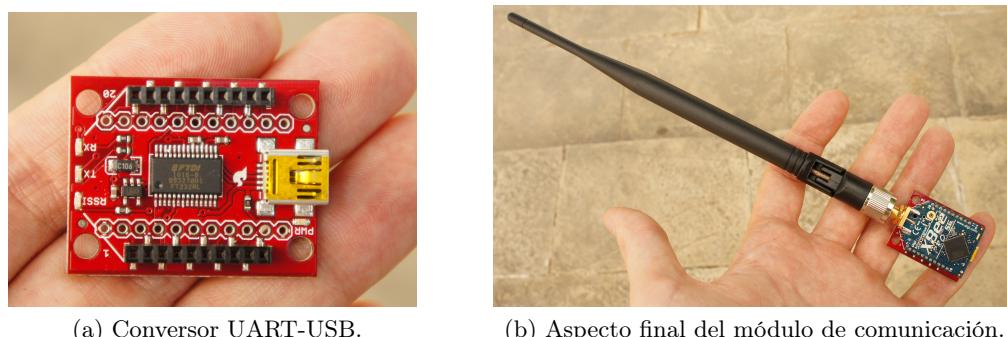


Figura 4.2: Conversor UART-USB y nodo final formado por el XBee-PRO 868, la antena y el conversor UART-USB.

En el Apéndice A se incluye el presupuesto del proyecto, donde puede observarse

un desglose detallado del equipo adquirido y los proveedores.

Toda la documentación de estos módulos se encuentra disponible gratuitamente en la página Web de Digi International<sup>1</sup>. En ella, se encontrarán las tablas de referencia de comandos, necesarias para la configuración de los módulos. En las próximas secciones se hará referencia a algunos parámetros de configuración, explicándolos brevemente. Para obtener una descripción más detallada de los mismos se remite al lector a la documentación indicada.

#### 4.2.2. Software X-CTU

X-CTU es una aplicación gráfica desarrollada y distribuida gratuitamente por Digi International. Este software permite interactuar, de una manera fácil e intuitiva, con el firmware utilizado en los módulos RF de dicha compañía. Concretamente, al ejecutar esta aplicación se muestra una ventana con cuatro pestañas que son:

- **PC Settings.** Permite al usuario seleccionar el puerto COM deseado y configurar este adecuadamente para comunicarse con el módulo radio.

En GNU/Linux es necesario realizar una operación previa para que el programa reconozca correctamente el puerto deseado. Tras ejecutar X-CTU con *Wine* el programa se instala en el directorio `~/.wine/drivec/Archivos de programa/Digi/XCTU`. Al conectar el conversor UART-USB el sistema lo reconoce como `/dev/ttyUSB0`. Para que X-CTU pueda ver dicho puerto es necesario crear un enlace suave en la carpeta `dosdevices` de Wine. El comando para ello es: `ln -s /dev/ttyUSB0 ~/.wine/dosdevice/com5`. Una vez realizado esto, únicamente hay que ejecutar X-CTU, en la pestaña de *PC Settings* pinchar en *User Com Ports*, escribir el puerto creado en la casilla *Com Port Number* y, finalmente, *Add*. (ver Figura 4.3a).

- **Range Test.** Permite al usuario realizar pruebas de alcance entre dos módulos. Como se observa en la Figura 4.3b, dispone de las siguientes opciones:

- Modificar el tamaño de los paquetes de datos para las pruebas.
- Seleccionar los valores mínimo y máximo permitido de retardo de paquetes.
- Fijar un número concreto de paquetes a enviar en la prueba. También permite la opción de detener el proceso en caso de que alguno se pierda.
- Elegir el timeout de espera de datos. Una vez superado este valor, se da el paquete por perdido.
- Medir el RSSI de cada paquete recibido.
- Calcular el PER (*Packet Error Rate*) del total de paquetes enviados en la prueba.

- **Terminal.** Permite acceder al puerto COM mediante una emulación de terminal. Así se puede acceder al firmware mediante la utilización de comandos AT (ver Figura 4.3c).
- **Modem Configuration.** Posibilita la configuración del módulo a través de un interfaz gráfico. Además, desde esta pestaña es posible cambiar entre las diferentes versiones del firmware (ver Figura 4.3d).

---

<sup>1</sup><http://www.digi.com/>

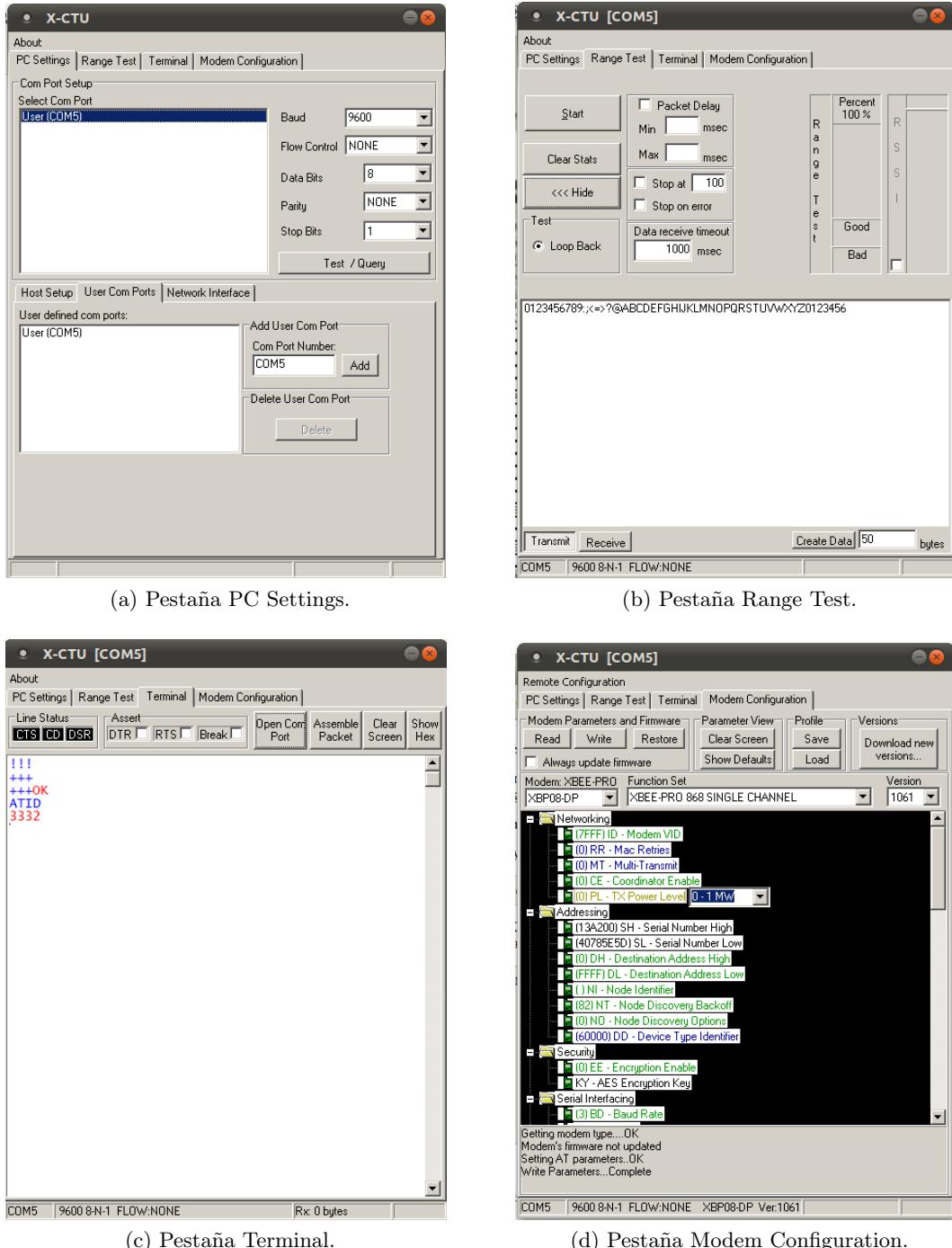


Figura 4.3: Pestañas del software X-CTU de Digi International.

#### 4.2.3. Primeras pruebas

Para verificar el correcto funcionamiento de los módulos XBee-PRO 868 adquiridos, se realizaron pruebas de transmisión y recepción con todos ellos. Los parámetros de configuración se mantuvieron prácticamente todos como venían por defecto, únicamente se modificaron los siguientes:

- **RR - MAC Retries.** Se le dió el valor 0. De esta manera cada paquete unicast

sólo se envía una única vez, aunque no se reciba su correspondiente acuse de recibo (ACK).

- **MT - Multiple Transmissions.** Se le dió el valor 0. De esta manera cada paquete broadcast sólo se envía una única vez.
- **PL - TX Power Level.** Se realizaron pruebas tanto con el valor 0, equivalente a 0 dBm, como con el valor 4, equivalente a la máxima potencia, 25 dBm.

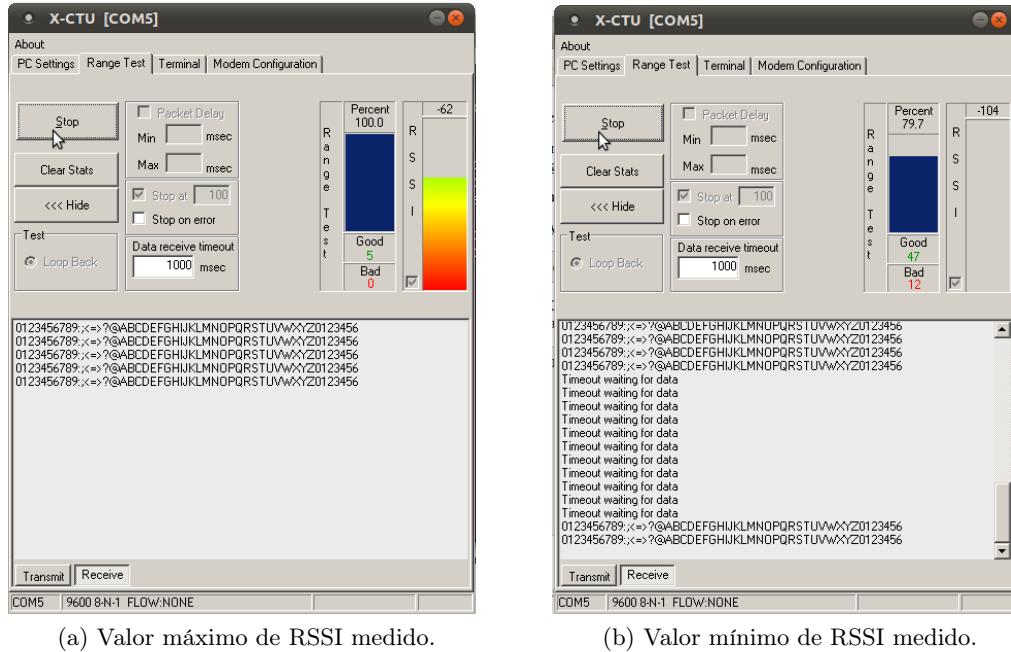
Para la realización de las pruebas con la aplicación *Range Test* se debe proceder de la siguiente manera. Un módulo debe configurarse en *loopback* o bucle, es decir, se conecta el pin DIN con el DOUT, de modo que dicho módulo retransmite lo que recibe. En la Figura 4.4 se puede observar como se implementó dicho *loopback* en un módulo utilizando un *jumper*.



Figura 4.4: Módulo configurable en *loopback* mediante un *jumper*.

De este modo, se verificó que los tres módulos adquiridos eran capaces de transmitir y recibir información correctamente entre ellos. Únicamente se destacan dos observaciones que se hicieron:

- Como muestra la Figura 4.5a, el valor máximo de RSSI medido es -62 dBm. Se averiguó, consultando en el foro oficial de Digi International, que estos módulos tienen la limitación de que únicamente muestran los valores de RSSI próximos al nivel de sensibilidad del mismo. Por ello, cuando los módulos se encuentren próximos el máximo valor de RSSI que se registrará será siempre -62 dBm.
- Como se observa en la Figura 4.5b, el valor mínimo de RSSI medido es -104 dBm. Se realizaron varias pruebas, con distintas potencias de transmisión, en las que se buscaba el límite de distancia en el que empezaban a llegar los paquetes con errores, o directamente ni se recibían. En ningún caso se pudo registrar un valor de RSSI inferior a -104 dBm. Este hecho es importante, ya que entre la especificación del fabricante y los valores medidos existe una diferencia de 8 dB, un valor bastante considerable. Sobre esta situación no se encontró nada en Internet. Puesto que no se recibe ningún paquete con una potencia inferior a -104 dBm, para el estudio teórico se tomará este valor como sensibilidad del módulo.

Figura 4.5: Valores de mínimo y máximo de RSSI medidos con la aplicación *Range Test*.

## 4.3. Estudio teórico de un enlace radioeléctrico en el mar

### 4.3.1. Constitución del modelo

Antes de efectuar balances de pérdidas y ganancias de potencia, y cálculos de niveles de señal en diferentes interfaces de un sistema, se propone seguidamente un modelo general simplificado de sistema de radiocomunicación, al que se referirán los diferentes conceptos relativos a potencias, ganancias y pérdidas. El modelo se representa en la Figura 4.6 y consta de los siguientes elementos e interfaces:

1. TX: Transmisor. Se conecta a la antena a través del interfaz AT, donde se descartan pérdidas por simplificar el modelo, y se centra la atención en cómo afecta el medio al radioenlace.
2. Antena de transmisión ideal. Termina en el interfaz virtual de antena isotrópica IT, por donde se accede al medio de propagación.
3. Antena de recepción ideal, a través de cuya interfaz virtual de antena isotrópica de recepción IR, entra la señal al sistema receptor.
4. RX: Receptor. Se acopla a la antena a través del interfaz AR donde, por los mismos motivos explicados anteriormente, se descartan pérdidas.

Para la formulación de las ecuaciones de balance y diagramas de niveles se aprovechará la ventaja de la representación logarítmica de las señales y de las pérdidas, por lo que todas estas magnitudes se expresarán en dB y sus derivados. En el modelo se definen las siguientes potencias:

- $P_t(\text{dBm})$ : potencia entregada por el transmisor a la antena.

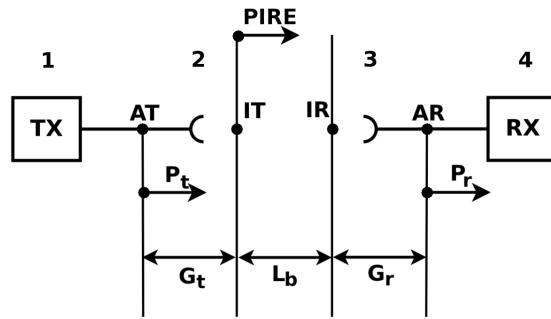


Figura 4.6: Modelo general simplificado de un sistema de radiocomunicación.

- $PIRE(dBm)$ : potencia isotrópica radiada equivalente en dirección hacia el receptor.
- $P_r(dBm)$ : potencia disponible a la entrada del receptor.

Únicamente se define la pérdida básica de propagación entre antenas isótropas,  $L_b$ . Se mide en  $dB$  y es función de: la frecuencia, la distancia, las alturas de las antenas, el modo y medio de propagación, etc.

Las ganancias que aparecen en el modelo son las correspondientes a las antenas:

- $G_t(dBi)$ : ganancia de potencia de la antena del transmisor.
- $G_r(dBi)$ : ganancia de potencia de la antena del receptor.

Para este estudio, de acuerdo con el equipo presentado en la Subsección 4.2.1, se utilizan los siguientes valores:

- $P_t = 25dBm$
- $G_t = 4,5dBi$
- $PIRE = P_t + G_t = 29,5dBm$
- $G_r = 4,5dBi$

#### 4.3.2. Balance de un enlace radioeléctrico

El balance del enlace es la relación que expresa la potencia disponible en el receptor en función de la potencia entregada por el transmisor y las diferentes pérdidas y ganancias que aparecen en el trayecto del transmisor al receptor. La ecuación general de balance del enlace es:

$$P_r = P_t + G_t - L_b + G_r \quad (4.1)$$

donde las pérdidas y las ganancias están en  $dB$ , y las potencias en unidades logarítmicas similares, que en este caso serán  $dBm$ .

Si esta potencia disponible en el receptor es superior a su sensibilidad, se considerará que existe comunicación entre el transmisor y el receptor. Se recuerda que en la

Subsección 4.2.3 se registró de manera empírica que el módulo XBee-PRO 868 tiene una sensibilidad de -104 dBm. Este valor será el que se empleará para determinar el alcance teórico del enlace.

#### 4.3.3. Enlace radioeléctrico en condiciones de espacio libre

En un sistema de radiocomunicaciones siempre interviene de alguna forma el medio de propagación. Sin embargo, se estudia el caso ideal de propagación en espacio libre como marco de referencia y para la determinación de la pérdida de propagación mínima que cabe esperar en el enlace.

Como se expuso en la Subsección 2.5.1, la pérdida básica de propagación en condiciones de espacio libre, expresada en dB, es:

$$L_{bf}[dB] = 32,45 + 20 \cdot \log(d[km]) + 20 \cdot \log(f[MHz]) \quad (4.2)$$

donde  $d[km]$  es la distancia en kilómetros entre el transmisor y el receptor,  $f[MHz]$  es la frecuencia de operación en megahercios.

En la Figura 4.7 se representa la pérdida básica de propagación en condiciones de espacio libre en función de la distancia entre el transmisor y el receptor, utilizando como frecuencia de trabajo  $f = 869,525MHz$ , que es la frecuencia central de la banda de operación.

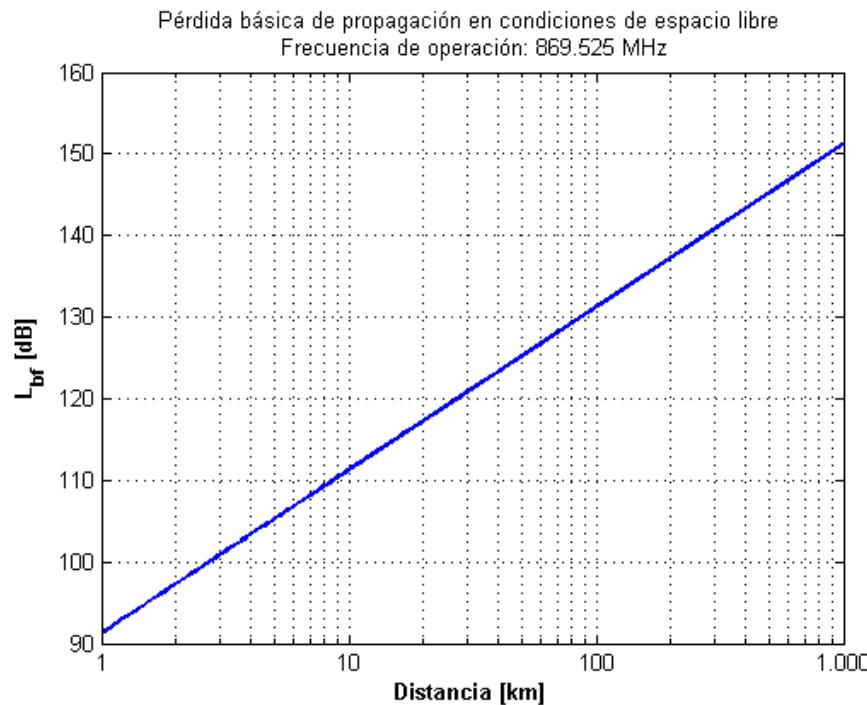


Figura 4.7: Pérdida básica de propagación en condiciones de espacio libre en función de la distancia, para una frecuencia de operación  $f = 869,525MHz$ .

Si evaluamos la ecuación general de balance del enlace, expresión 4.1, para una potencia entregada por el transmisor de 25 dBm, unas ganancias de transmisión y recepción de 4.5 dBi, la pérdida básica de propagación en el espacio libre para  $f = 869,525MHz$  y una sensibilidad del receptor de -104 dBm, la comunicación sería posible hasta una distancia de 218 kilómetros, como se muestra en la Figura 4.8.

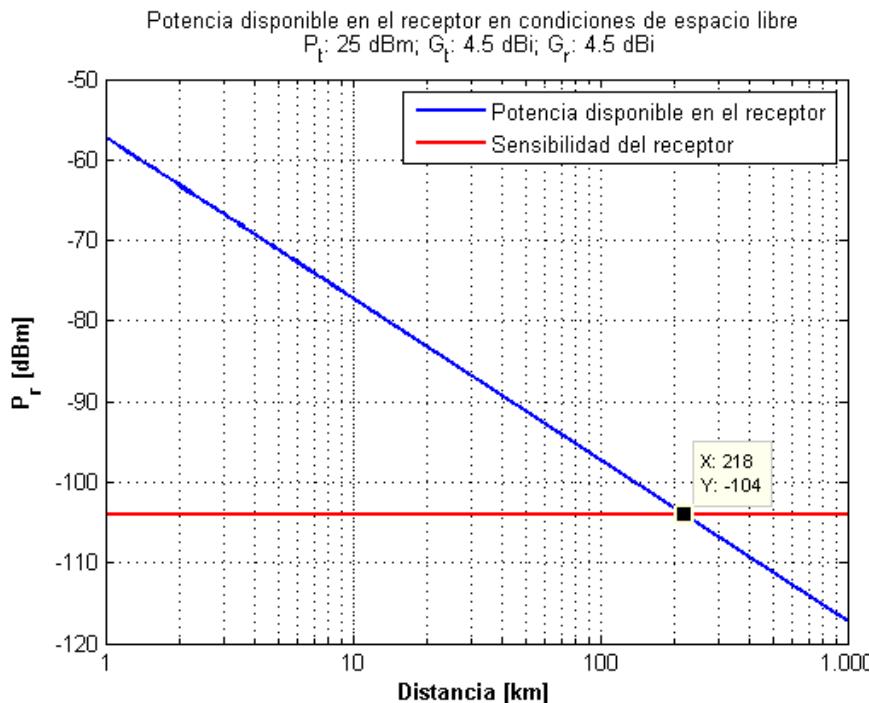


Figura 4.8: Potencia disponible a la entrada del receptor en función de la distancia en condiciones de espacio libre, para una frecuencia de operación  $f = 869,525 MHz$ .

Como se indicó al principio, este resultado se corresponde con el caso ideal, sin tener en cuenta de ninguna forma el medio de propagación. Por ello, en una implementación real no se espera lograr estos alcances. A continuación, se presenta otro modelo para predecir de una forma más realista las pérdidas básicas de propagación de una onda que viaja del transmisor al receptor sobre la superficie del mar.

#### 4.3.4. Influencia de la superficie sobre la que discurre la onda

Cuando el transmisor y el receptor se encuentran situados sobre la superficie terrestre, y entre ambos existe una visibilidad directa, se modela la propagación mediante un rayo directo y uno reflejado en la superficie. Además, dependiendo de algunos factores como: la naturaleza del terreno, la frecuencia y la polarización de la onda, puede existir también una componente de onda de superficie.

La expresión general del campo recibido, en estas condiciones viene dada mediante la llamada *ecuación general de la propagación*:

$$e = e_0[1 + R \cdot e^{-j\Delta} + (1 - R) \cdot A \cdot e^{-j\Delta}] \quad (4.3)$$

donde:

- $e$ : intensidad de campo en recepción en las condiciones reales.
- $e_0$ : intensidad de campo en condiciones de espacio libre.
- $\Delta$ : ángulo de desfasamiento entre la componente directa y la reflejada.
- $R$ : coeficiente de reflexión de la superficie.

- A: término de atenuación de la onda de superficie.

En esta expresión, el primer término del paréntesis equivale a la componente de onda directa, el segundo a la de onda reflejada en el suelo, y el tercero a la onda de superficie. El ángulo  $\Delta$  es:

$$\Delta = \frac{2\pi\Delta l}{\lambda} \quad (4.4)$$

donde  $\Delta l$  es la diferencia entre el recorrido del rayo reflejado y el del rayo directo, y  $\lambda$  es la longitud de onda.

El coeficiente de reflexión complejo  $R$ , se especifica en función de su módulo y fase:

$$R = |R|e^{-j\beta} \quad (4.5)$$

Tanto  $|R|$  como  $\beta$  son función de la frecuencia, polarización, características eléctricas del suelo y ángulo de incidencia  $\Psi$ .

Para el estudio de la propagación radioeléctrica, se caracteriza el terreno por los parámetros eléctricos: constante dieléctrica relativa,  $\varepsilon_r$ , y conductividad,  $\sigma$ (mhos/m). En la Recomendación UIT-R P.527, se facilitan datos sobre estos parámetros. Concretamente, para el agua salada (salinidad media), a una temperatura de 20°C, y a una frecuencia de 869.525 MHz, los valores son:  $\varepsilon_r = 70$  y  $\sigma = 5$  S/m.

Se define la permitividad compleja del suelo como:

$$\varepsilon_0 = \varepsilon_r - j60\sigma\lambda \quad (4.6)$$

A partir de este parámetro, se define el parámetro  $z$  en función de la polarización de la onda, y del ángulo de incidencia  $\Psi$ , del modo siguiente:

*Polarización vertical:*

$$z = \frac{[\varepsilon_0 - \cos^2 \Psi]^{1/2}}{\varepsilon_0} \quad (4.7)$$

*Polarización horizontal:*

$$z = [\varepsilon_0 - \cos^2 \Psi]^{1/2} \quad (4.8)$$

El coeficiente de reflexión  $R$  en una superficie plana viene dado por:

$$R = \frac{\sin \Psi - z}{\sin \Psi + z} \quad (4.9)$$

Por lo que resulta:

*Polarización vertical:*

$$R_V = \frac{\varepsilon_0 \sin \Psi - \sqrt{\varepsilon_0 - \cos^2 \Psi}}{\varepsilon_0 \sin \Psi + \sqrt{\varepsilon_0 - \cos^2 \Psi}} \quad (4.10)$$

*Polarización horizontal:*

$$R_H = \frac{\sin \Psi - \sqrt{\varepsilon_0 - \cos^2 \Psi}}{\sin \Psi + \sqrt{\varepsilon_0 - \cos^2 \Psi}} \quad (4.11)$$

Para este estudio, la antena dipolo  $\lambda/2$  es perpendicular al plano de la tierra, la componente eléctrica de la onda es emitida perpendicularmente a dicho plano y, por lo tanto, se trabajará con polarización vertical. Además, el enlace transcurrirá sobre la superficie del mar que, por su salinidad, es muy conductiva. En estas condiciones, y dado que la altura de las antenas generalmente será muy inferior a la distancia entre el transmisor y el receptor, la incidencia será casi rasante,  $\Psi \approx 0$ , y resulta  $|R| \approx 1$  y  $\beta = \pi$ . Esto hará que si una onda polarizada verticalmente se refleja sobre dicha superficie, sufra un cambio de fase de 180 grados, o  $\pi$  radianes.

#### 4.3.5. Modelo de propagación de tierra plana

El modelo de tierra plana es aplicable a distancias cortas, para las que puede despreciarse la curvatura terrestre, y con terreno liso.

Al operar a una frecuencia superior a 150 MHz, el efecto de onda de superficie puede despreciarse.

En la Figura 4.9 se muestra el esquema de rayos de este modelo y los parámetros asociados:

- $T$ : antena transmisora.
- $T'$ : reflejo geométrico de la antena transmisora.
- $R$ : antena receptora.
- $h_t$ : altura de la antena transmisora.
- $h'_t$ : altura de la antena transmisora reflejada. Es igual que  $h_t$ .
- $h_r$ : altura de la antena receptora.
- $d$ : distancia entre el transmisor y el receptor.
- $\Psi$ : ángulo de incidencia del rayo reflejado.
- $RD$ : rayo directo.
- $RR$ : rayo reflejado.

Mediante geometría básica se obtiene:

*Ángulo de incidencia:*

$$\Psi = \arctan \left( \frac{h_t + h_r}{d} \right) \quad (4.12)$$

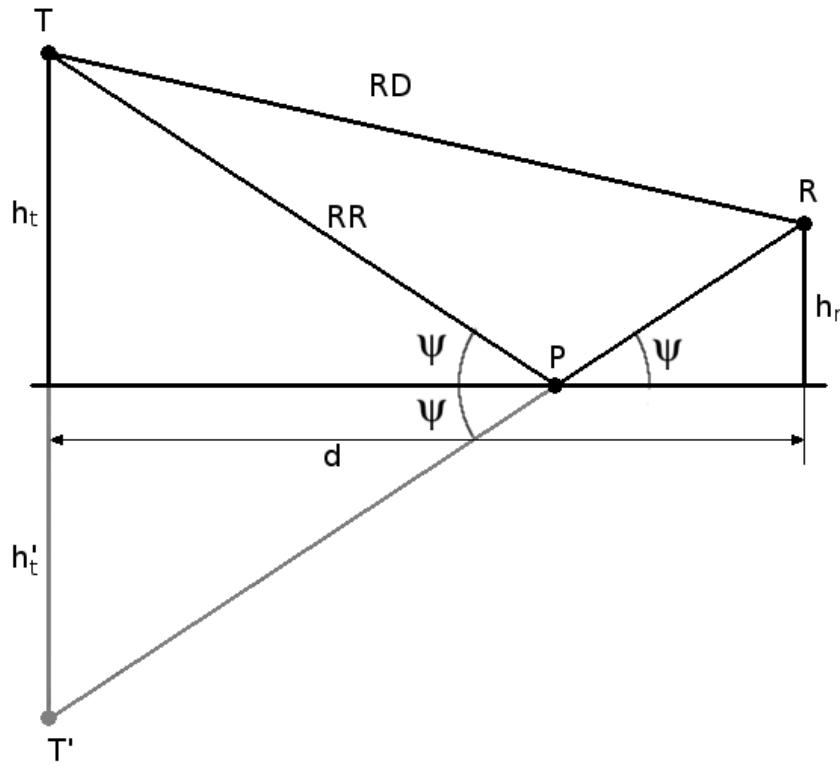


Figura 4.9: Esquema de rayos del modelo de propagación de tierra plana.

*Diferencia de trayectos:*

$$\Delta l = TPR - TR = \sqrt{d^2 + (h_t + h_r)^2} - \sqrt{d^2 + (h_t - h_r)^2} \approx \frac{2h_t h_r}{d} \quad (4.13)$$

*Diferencia de fase:*

$$\Delta = \frac{4\pi h_t h_r}{\lambda d} \quad (4.14)$$

Si llamamos  $e$  al módulo del campo, aplicando la ecuación 4.3, prescindiendo de su componente de onda de superficie, se tiene:

$$e = e_0 \left| \left\{ 1 + |R| \cdot e^{-j(\Delta + \beta)} \right\} \right| = e_0 \sqrt{1 + |R|^2 + 2|R| \cos(\Delta + \beta)} \quad (4.15)$$

La expresión 4.14 proporciona el valor de  $\Delta$ , y de las relaciones 4.10 y 4.11 se obtienen  $|R|$  y  $\beta$ .

La pérdida básica de propagación será:

$$l_b = \left( \frac{4\pi d}{\lambda} \right)^2 \cdot \left( \frac{e_0}{e} \right)^2 = \frac{\left( \frac{4\pi d}{\lambda} \right)^2}{1 + |R|^2 + 2|R| \cos(\Delta + \beta)} \quad (4.16)$$

Como, en general,  $\Psi = 0$ , dado que  $h_t$  y  $h_r$  son muy inferiores a  $d$ , resulta  $|R| \approx 1$  y  $\beta = \pi$ , por lo que:

$$e = e_0 \sqrt{2(1 - \cos \Delta)} = 2e_0 \left| \sin \frac{\Delta}{2} \right| = 2e_0 \left| \sin \frac{2\pi h_t h_r}{\lambda d} \right| \quad (4.17)$$

y

$$l_b = \frac{\left(\frac{4\pi d}{\lambda}\right)^2}{2(1 - \cos \Delta)} \quad (4.18)$$

Del análisis de estas ecuaciones, 4.17 y 4.18, se deduce que variarán de forma oscilatoria alrededor del valor en el espacio libre,  $e_0$  y  $l_{bf}$  respectivamente, generándose los denominados *lóbulos de interferencia* entre el rayo directo y el rayo reflejado. Estas variaciones reflejan el hecho físico del refuerzo o cancelación del campo, según las ondas incidente y reflejada se sumen en concordancia o en oposición de fase.

En la Figura 4.10 se representa la pérdida básica de propagación descritas por este modelo, ecuación 4.18, en función de la distancia, para una frecuencia  $f = 869,525\text{MHz}$  y unas alturas de antena  $h_t = h_r = 2$  metros. Además, se incluye la pérdida básica de propagación en condiciones de espacio libre, pudiéndose observar la oscilación ya mencionada de la primera alrededor del valor en el espacio libre, hasta el punto  $d = \frac{12h_th_r}{\lambda}$ .

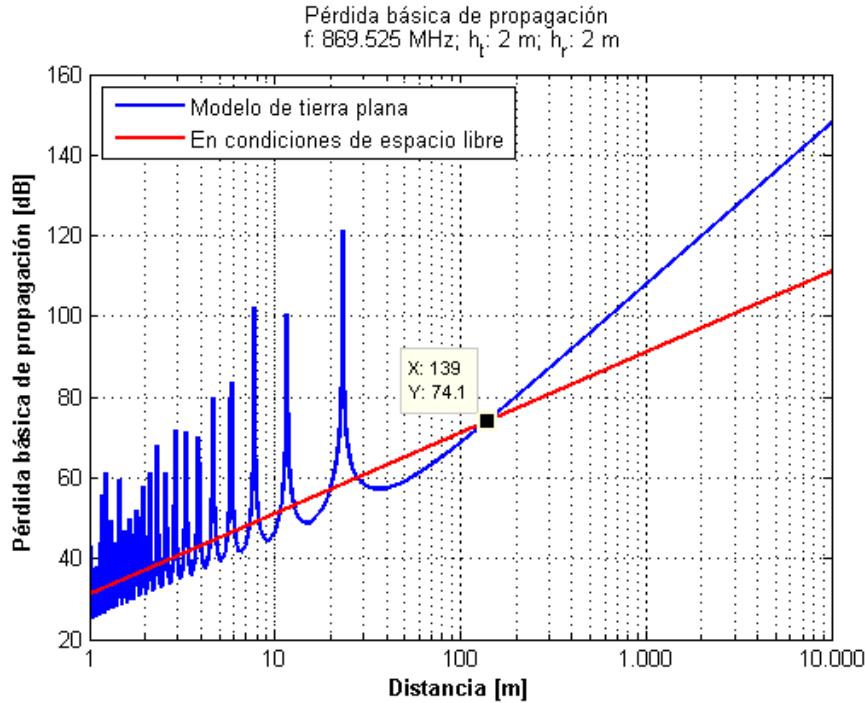


Figura 4.10: Pérdida básica de propagación descrita por el modelo de tierra plana.

Los máximos de esta función se producen cuando el  $\cos\Delta$  vale 1, anulando el denominador de la función, y haciéndola tender a infinito. Esto ocurre cuando:

$$\Delta = 2n\pi \implies d = \frac{2h_th_r}{n\lambda} \quad (4.19)$$

Los mínimos se producen cuando el  $\cos\Delta$  toma el valor -1, haciendo máximo el denominador de la función. Esto sucede cuando:

$$\Delta = (2n + 1)\pi \implies d = \frac{4h_th_r}{\lambda(2n + 1)} \quad (4.20)$$

Para  $d = \frac{12h_th_r}{\lambda}$ ,  $e = e_0$  y para distancias mayores  $e$  es ya siempre menor a  $e_0$ . Esta es la zona habitual de trabajo en los modelos de tierra plana. En ella el argumento del seno en 4.17 es muy pequeño, por lo que puede ponerse:

$$e = e_0 \frac{4\pi h_th_r}{\lambda d} \quad (4.21)$$

que es la fórmula habitualmente utilizada para la propagación en condiciones de tierra plana.

En este caso, la pérdida básica de propagación será:

$$l_b = \frac{d^4}{(h_t h_r)^2} \quad (4.22)$$

y en dB:

$$L_b = 40 \cdot \log(d[km]) - 20 \cdot \log(h_t \cdot h_r) + 120 \quad (4.23)$$

Es decir, resulta independiente de la frecuencia de la onda y decrece con la distancia a la cuarta, en vez de al cuadrado como sucede en condiciones de espacio libre.

En la Figura 4.11 se representan en función de la distancia, y utilizando la misma frecuencia y alturas que en el caso anterior:

- La pérdida básica de propagación en condiciones de estado libre, ecuación 4.2.
- La pérdida básica de propagación del modelo de tierra plana, ecuación 4.18 en unidades logarítmicas.
- La pérdida básica de propagación del modelo de tierra plana en su forma simplificada para distancias  $d \geq \frac{12h_t h_r}{\lambda}$ , ecuación 4.23. En este caso  $d \geq 139,124$  metros.

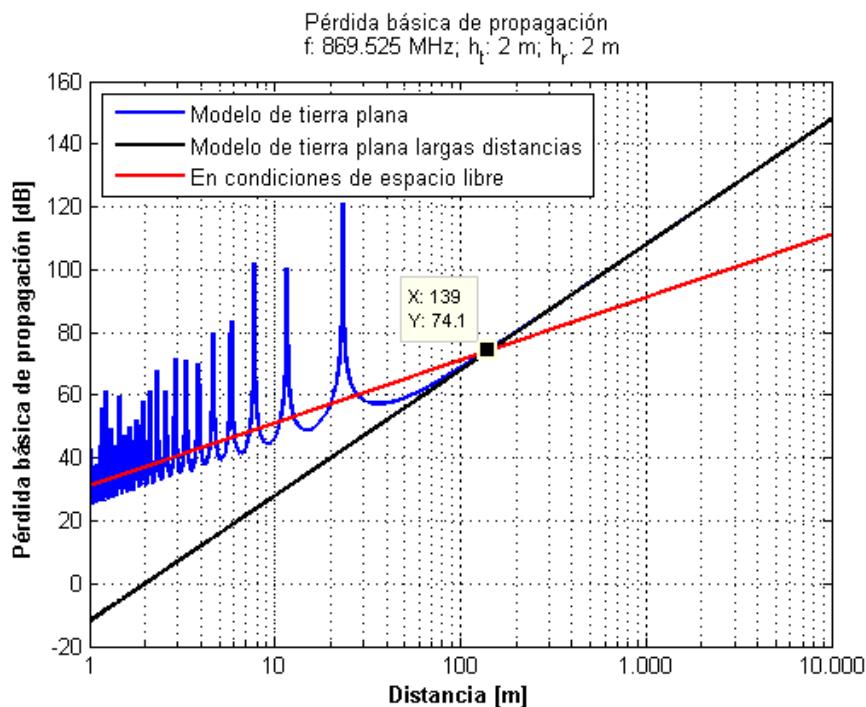


Figura 4.11: Pérdida básica de propagación de varios modelos.

En la Figura 4.12 se evalúa la ecuación general de balance del enlace, expresión 4.1, utilizando las pérdidas básicas de propagación descritas por el modelo de tierra plana, expresión 4.18, y los valores:  $f = 869,525 MHz$ ,  $P_t = 25 dBm$ ,  $G_t = G_r = 4,5 dBi$  y  $h_t = h_r = 2$  metros.

Se observa que, en este caso, la distancia máxima a la que se podría lograr comunicación entre el transmisor y el receptor es de 5.6 kilómetros, menos de un 3 % del valor

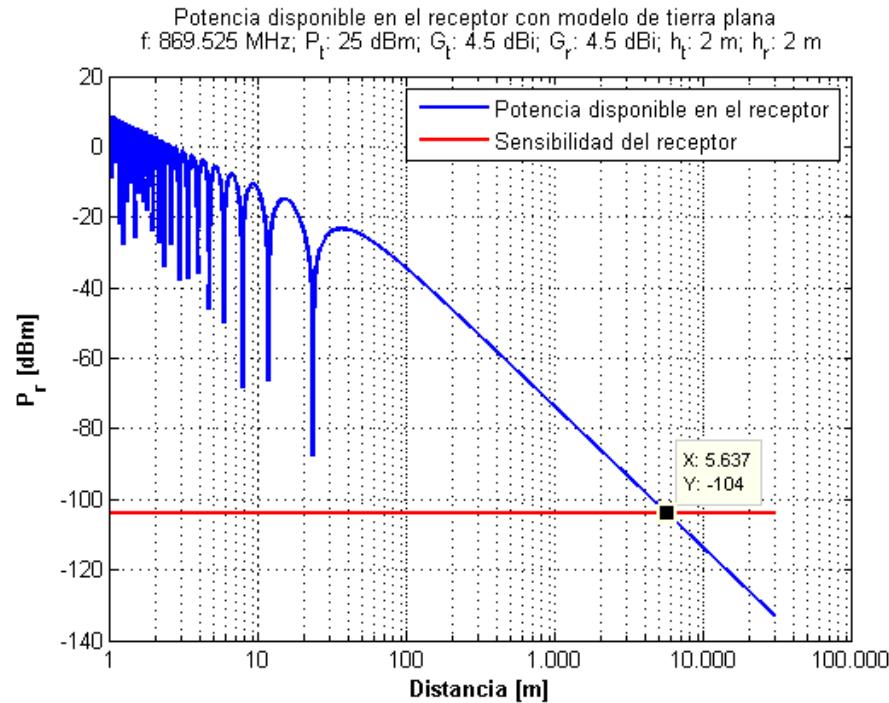


Figura 4.12: Potencia disponible a la entrada del receptor en función de la distancia utilizando el modelo de tierra plana.

ideal calculado en condiciones de espacio libre. Aún así, este alcance obtenido sigue siendo un resultado teórico, en el que se supone el mar como una superficie lisa y no se contemplan otros factores que estarán presentes en el caso real, como: pérdidas de potencia en los equipos, variación de la altura relativa de las antenas, difracción por barcos u el propio oleaje, atenuación por gases o partículas en suspensión, etc.

Para profundizar en estos temas, se remite al lector al libro de *Transmisión por radio* (Hernando-Rábanos, 2008). El código utilizado para las simulaciones en MATLAB se incluye en el Apéndice B. También se propone la utilización de herramientas libres para el cálculo de enlaces de comunicación, como: *Radio Mobile*<sup>2</sup> o *SPLAT!*<sup>3</sup>.

## 4.4. Implementación de un enlace simple de datos en el mar

Con el objetivo de determinar la distancia máxima a la que podrán situarse las boyas entre sí para formar la red ad hoc, se han realizado una serie de medidas de campo, que se detallan a continuación.

### 4.4.1. Escenario

Se han implementado tres enlaces simples de datos. En todos los casos, tanto el nodo transmisor como el receptor se encontraban en tierra firme, pero situados de modo que prácticamente la totalidad del vano del enlace discurría sobre la superficie del mar. Concretamente, como se puede observar en la Figura 4.13, uno de ellos estaba

<sup>2</sup><http://www.cplus.org/rmw/>

<sup>3</sup><http://www.qsl.net/kd2bd/splat.html>

ubicado de manera permanente al final del espigón del Puerto Pesquero de Vilanova i la Geltrú, y el otro se situó en diferentes puntos de la costa logrando enlaces aproximados de: 2, 3 y 3.5 kilómetros.

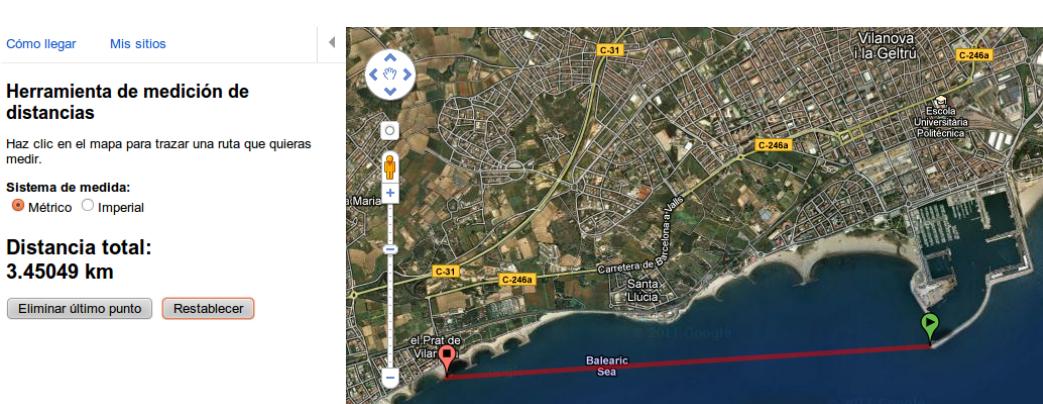
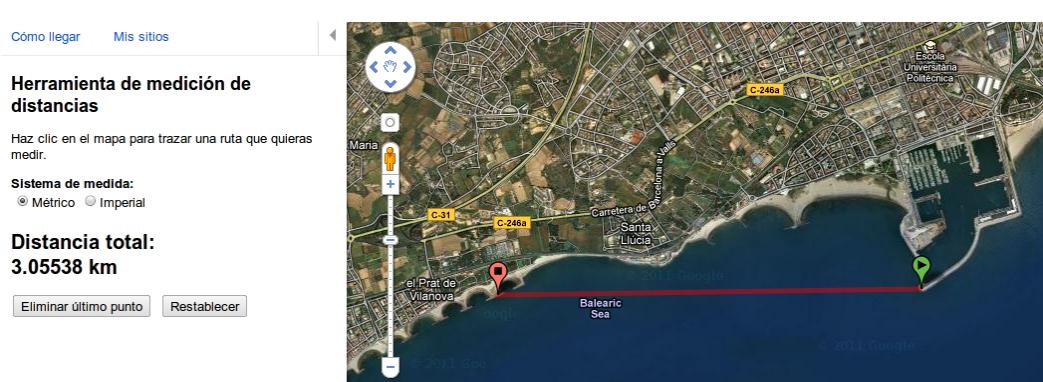
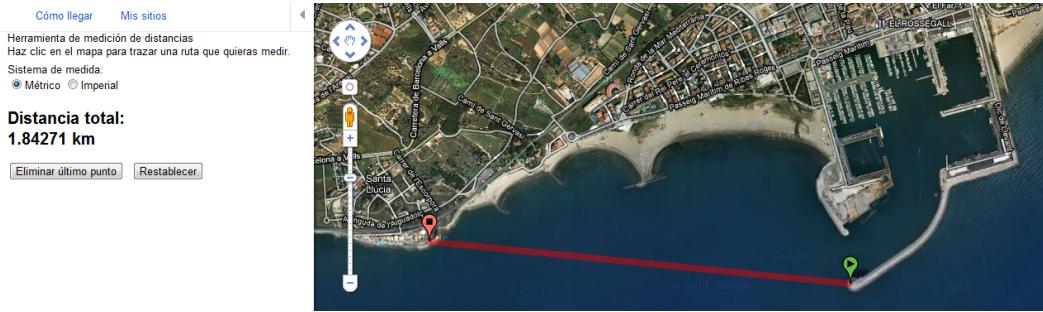


Figura 4.13: Enlaces implementados en las medidas de campo.

En la Tabla 4.2 se presentan las coordenadas geográficas y altura aproximada de cada uno de los puntos.

Número de enlace	Posición del nodo A	Altura aprox. del nodo A	Posición del nodo B	Altura aprox. del nodo B	Distancia (km)
<b>Enlace 1</b>	41.206017 N 1.729322 E	2.5 m	41.207619 N 1.70717 E	5 m	1.84
<b>Enlace 2</b>	41.206017 N 1.729322 E	2.5 m	41.205603 N 1.693043 E	2.5 m	3.06
<b>Enlace 3</b>	41.206017 N 1.729322 E	2.5 m	41.204021 N 1.688182 E	2.5 m	3.45

Tabla 4.2: Posición y altura de los nodos en la implementación de los enlaces simples de datos.

#### 4.4.2. Procedimiento

Se han empleado, para realizar las distintas medidas, dos módulos XBee-PRO 868 con su configuración por defecto de fábrica, salvo los siguientes parámetros:

- **RR - MAC Retries.** Tiene el valor 0. De esta manera cada paquete unicast sólo se envía una única vez, aunque no se reciba su correspondiente acuse de recibo (ACK).
- **MT - Multiple Transmissions.** Con el valor 0. De esta manera cada paquete broadcast sólo se envía una única vez.
- **PL - TX Power Level.** Utilizando el valor 4, equivalente a la máxima potencia, 25 dBm (315 mW).

El nodo que realiza las medidas, al que se llamará nodo A, consiste en un ordenador portátil con el software X-CTU y un módulo XBee-PRO 868 conectado a él por el puerto USB. El nodo B, configurado en *loopback*, utiliza únicamente una batería con salida USB para su alimentación.

Un paquete recibido con éxito implica que:

1. El nodo A lo envía, y llega correctamente al B.
2. El nodo B, al estar configurado en *loopback*, retransmite el mismo paquete, y el A lo recibe correctamente, anotándolo como bueno.

Por lo tanto, cada paquete recibido por el nodo A recorre dos veces el enlace, una de ida y otra de vuelta. Más adelante, se profundizará en este hecho.

Concretamente, utilizando la herramienta *Range Test*, incluida en el software X-CTU de Digi International, se ha medido:

- El porcentaje de paquetes recibidos con éxito. Para ello, se envían desde el nodo A tandas de 100 paquetes, midiendo el número de ellos que regresan correctamente. Se han empleado paquetes de 50 bytes de tamaño, y un timeout de recepción de paquete de 1.5 segundos, en el nodo A.
- El nivel de potencia de señal recibida o RSSI. Para ello, se selecciona la opción en el software y se envían paquetes anotando, en caso de que se reciban con éxito, su nivel de RSSI.

Número de enlace	Jornada	Porcentaje de paquetes recibidos correctamente			RSSI (dBm)		
		Min.	Media	Max.	Min.	Media	Max.
<b>Enlace 1 (2 km)</b>	Jornada 1	83 %	92 %	100 %	-103	-96.8	-93
	Jornada 2	82 %	91.4 %	98 %	-102	-96.8	-92
<b>Enlace 2 (3 km)</b>	Jornada 1	-	-	-	-	-	-
	Jornada 2	-	-	-	-	-	-
<b>Enlace 3 (3.5 km)</b>	Jornada 1	-	-	-	-	-	-
	Jornada 2	-	-	-	-	-	-

Tabla 4.3: Medidas de porcentaje de paquetes recibidos correctamente y RSSI realizadas para varios enlaces sobre el mar.

Cada una de estas dos medidas se ha repetido 5 veces en cada uno de los 3 enlaces. Además, se han realizado dos jornadas de medida, una el día 29 de Junio y la otra el 13 de Julio de 2011. Ambos días, el mar estaba tranquilo, con un oleaje débil, temperatura del agua de 22-25°C, temperatura ambiente 26-28°C y humedad aproximada del 75 %.

#### 4.4.3. Medidas

Para las medidas se empleó el equipo ya mencionado, con:  $P_t = 25dBm$ ,  $G_t = 4,5dBi$ ,  $G_r = 4,5dBi$ . Las alturas de las antenas se indican en la Tabla 4.2. En la Figura 4.14 se muestra una imagen de cada uno de los nodos en su posición.

En la Tabla 4.3 se muestran los resultados de las medidas realizadas. Concretamente de las 5 anotadas en cada caso se indican: los valores mayor y menor, y la media de todas.

Como se observa, únicamente se consiguió comunicación en el primer enlace, con una distancia del enlace de 1.84 kilómetros. En los demás casos no se registró ningún paquete en el nodo A, a pesar de realizar numeroso intentos: desplazando el nodo B, aproximándolo a la orilla, elevándolo, etc.



(a) Nodo A.



(b) Nodo B en el enlace 1.



(c) Nodo B en el enlace 2.



(d) Nodo B en el enlace 3.

Figura 4.14: Nodos en los distintos enlaces.

#### 4.4.4. Análisis de probabilidad

Una de las opciones que permite el firmware de los módulos XBee-PRO 868 es determinar el número de reintentos de envío de paquete a nivel de MAC. Concretamente, se pueden definir en los parámetros de configuración *MAC Retries (RR)* para paquetes unicast, y *Multiple Transmissions (MT)* para broadcast. En las pruebas realizadas estos valores se encontraban a 0, es decir, cada paquete se enviaba una única vez, llegase o no con éxito.

De la Tabla 4.3 se puede determinar que, para el primer enlace, la probabilidad de recibir un paquete correctamente es aproximadamente del 92 %. Como ya se comentó anteriormente, por el método de medida utilizado, esto implica que el paquete haya recorrido dos veces el enlace, una de ida ( $A \rightarrow B$ ) y otra de vuelta ( $B \rightarrow A$ ). Por lo tanto, si se define:

- $P_{AA}$ : probabilidad de que un paquete realice la ida y la vuelta correctamente. Este sería el valor medido con el procedimiento utilizado, que para el enlace 1 es el 92 %.
- $P_{AB}$ : probabilidad de que un paquete del nodo A sea recibido con éxito por el B.
- $P_{BA}$ : probabilidad de que un paquete del nodo B sea recibido con éxito por el A.

, la relación entre ellas será:

$$P_{AA} = P_{AB} \cdot P_{BA} \quad (4.24)$$

Si se considera  $P_{AB} = P_{BA}$ , entonces:

$$P_{AA} = P_{AB}^2 \quad (4.25)$$

, por lo tanto:

$$P_{AB} = P_{BA} = \sqrt{P_{AA}} = 96 \% \quad (4.26)$$

Si se quiere garantizar que un paquete del nodo A al B llegue correctamente con una probabilidad de éxito,  $P_E \geq 99 \%$ , sea  $n$  el número de reintentos a nivel de MAC de envío del paquete, entonces:

$$P_E = 1 - (1 - P_{AB})^n \quad (4.27)$$

Por ello, el número de reintentos  $n$  será el entero superior del resultado de esta expresión:

$$n = \frac{\log_{10}(1 - P_E)}{\log_{10}(1 - P_{AB})} = \frac{\log_{10}(1 - P_E)}{\log_{10}(1 - \sqrt{P_{AA}})} = 1,43 \rightarrow 2 \quad (4.28)$$

Y de esta manera:

$$P_E = 1 - (1 - P_{AB})^n = 1 - (1 - 0,96)^2 = 0,9984 \rightarrow 99,84 \% \quad (4.29)$$

#### 4.5. Discusión de resultados

En la Figura 4.15 se muestran las medidas realizadas en comparación con el caso teórico del modelo de tierra plana. Se observa que la mayoría de ellas se encuentran

a más de 10 dB por debajo del valor teórico. Además, las diferencias entre el valor máximo y mínimo medidos en un mismo punto son de hasta 10 dB. Como se comentó en el Capítulo 2, existen casos similares en los que para distancias superiores a 750 metros estas variaciones alcanzaban hasta 25 dB (Fitz et al., 2005).

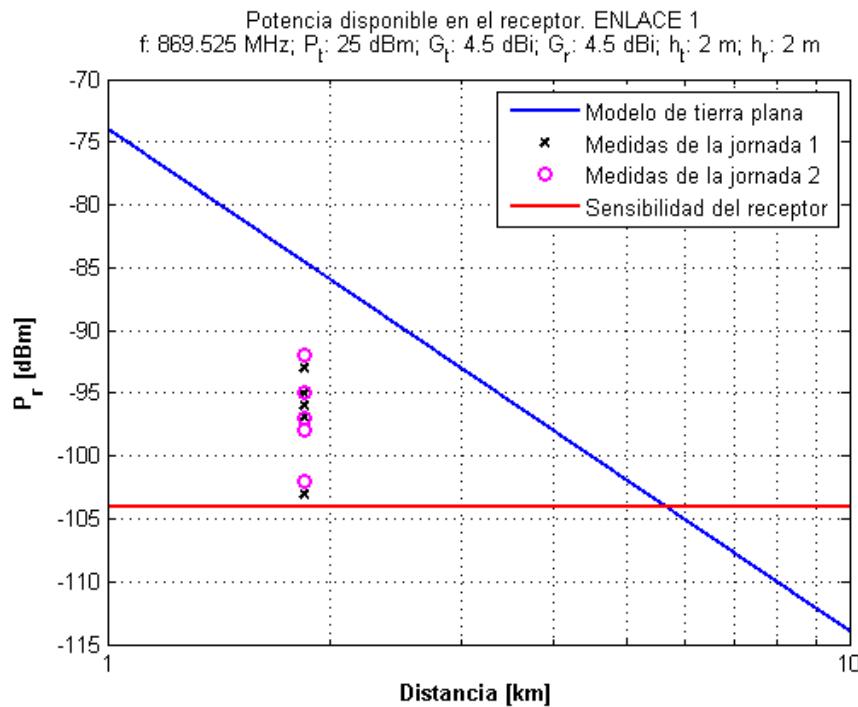


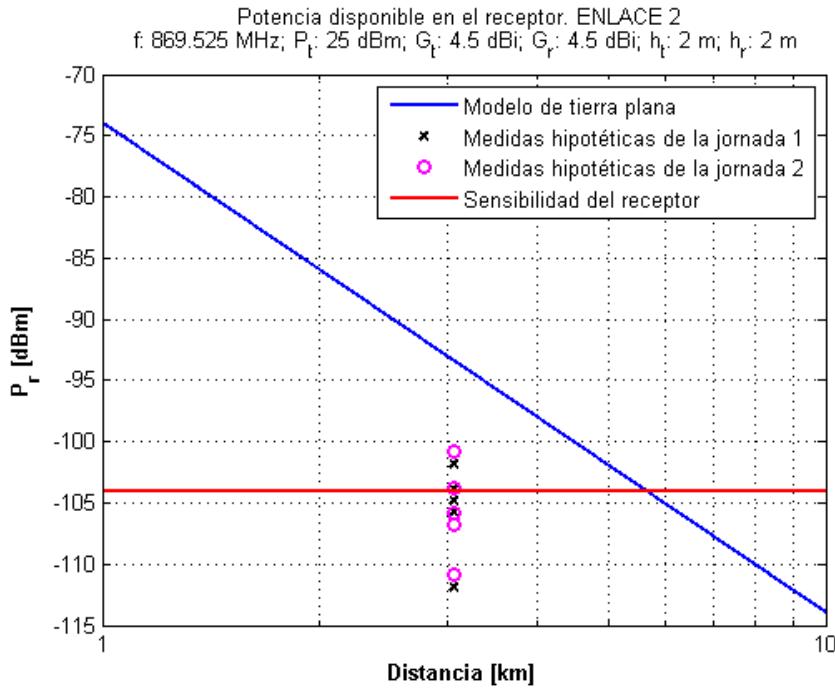
Figura 4.15: Medidas realizadas de potencia disponible en el receptor para enlace 1.

Como se reseñó en la Subsección 4.3.5, el modelo de tierra plana supone el mar como una superficie lisa y no contempla otros factores que están presentes en el caso real, como: pérdidas de potencia en los equipos, difracción por barcos u el propio oleaje, atenuación por gases o partículas en suspensión, etc. Estos efectos no contemplados, en base a las medidas realizadas, se pueden cuantificar en una reducción de la potencia recibida de entre 8 y 19 dB, valores muy considerables.

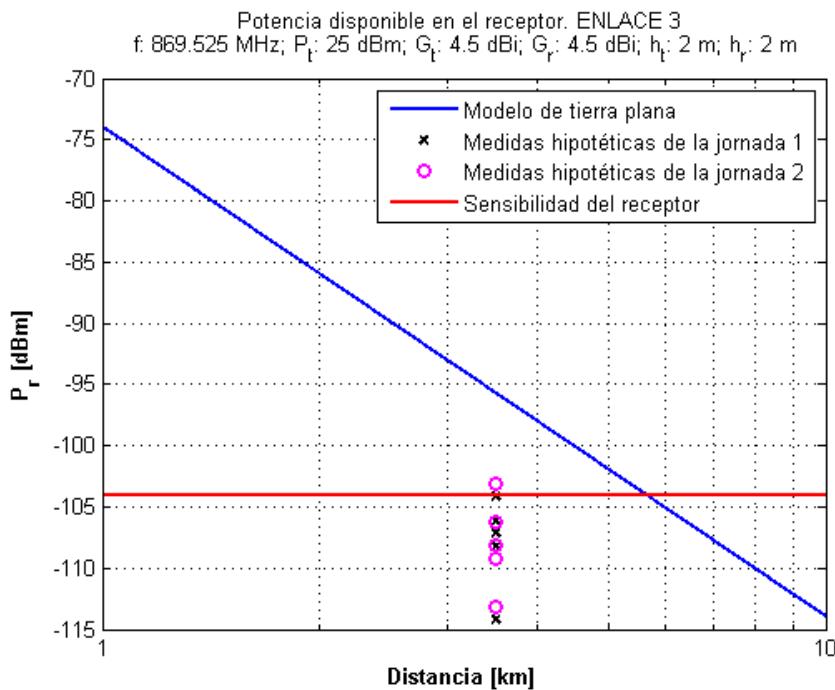
Si se desplazan las medidas realizadas a las distancias equivalentes de los enlaces 2 y 3, manteniendo siempre la misma diferencia con el caso teórico (ver Figura 4.16), se observa que algunos valores todavía se encuentran por encima de la sensibilidad del receptor. Por lo tanto, se tendría que haber registrado algún paquete como recibido, pero esto no fué así en ningún caso. Por ello, se afirma que la reducción de entre 8 y 19 dB sobre la potencia disponible en el receptor teórica aumenta con la distancia; siendo en el enlace 2, de 3 kilómetros, superior a 11 dB en todas las medidas. Este efecto también puede atribuirse a la variación de la altura de la antena del nodo B, que en los enlaces 2 y 3 es menor que en el primero.

Se recuerda que existen experiencias publicadas en las que, con los mismos módulos XBee-PRO 868 situados en boyas a una altura de 1.5 m sobre el nivel del mar, se llegaron a implementar enlaces de hasta 3 kilómetros. (Sieber et al., 2009)

Por otra parte, Garropo et al. (2009) realizan una campaña extensa de medidas en entorno marino y sus resultados se ajustan al caso teórico del modelo de tierra plana con una diferencia menor de 2 dB.



(a) Medidas hipotéticas en el enlace 2.



(b) Medidas hipotéticas en el enlace 3.

Figura 4.16: Medidas hipotéticas de potencia recibida en los enlaces 2 y 3.

En base a lo comentado, se propone repetir las medidas sobre tierra firme para tratar de determinar el efecto que produce el mar. Además, también sería conveniente realizar una campaña de medidas empleando una embarcación, pudiendo registrar datos durante todo el trayecto hasta la pérdida de comunicación entre nodos. Por último, se propone modificar el número de repeticiones a nivel de mac, acorde a la

Subsección 4.4.4, y determinar así si el análisis probabilístico realizado se ve reflejado, o no, en la realidad.



# Capítulo 5

## Software

*No fracasé, sólo descubrí 999 maneras de cómo no hacer una bombilla.*

Thomas Alva Edison

**RESUMEN:** En este capítulo se describe el software desarrollado para el módulo de entrega remota de datos del sistema LIDO DCL. Concretamente, este módulo se empleará en la implementación de dicho sistema sobre boyas autónomas en el mar, donde cada una actuará como un nodo dentro de una red ad hoc de datos. En primer lugar, se explica la funcionalidad de dicho módulo. A continuación, se realiza un análisis cualitativo de los tipos de protocolos de encaminamiento existentes para esta clase de redes, centrándose en el AODV y versiones basadas en él. Se presenta el AODV-LAB, una nueva implementación del AODV diseñada para este proyecto. Por último, se detalla el desarrollo en C/Linux del módulo de entrega remota de datos, se simula el mismo para validar las funcionalidades deseadas y se comentan los resultados obtenidos.

### 5.1. Módulo de entrega remota de datos

Como ya se vió en la Sección 1.2.2, el sistema LIDO DCL está orientado al estudio y mitigación del impacto ambiental de la contaminación acústica en el medio marino. Se encarga de la aquisición de datos acústicos mediante hidrófonos para, a continuación, realizar el procesado de los mismos y, finalmente, en caso de detectar algún evento acústico definido inicialmente, enviar una alarma con dicha información a un nodo colector.

La primera parte del sistema, correspondiente a las etapas de adquisición, procesado de audio y generación de alarmas, ya ha sido desarrollada y probada en el LAB. La idea de este proyecto ha sido, por lo tanto, trabajar a partir ahí para desarrollar finalmente un prototipo del sistema completo capaz de implementarse sobre boyas autónomas. Concretamente, el software implementado se encarga de:

- Detectar la aparición de nuevas alarmas en el sistema.
- Descubrir rutas hacia el nodo colector empleando un protocolo de encaminamiento.

- Envío de las alarmas a través de dichas rutas.
- Verificar la correcta entrega de las mismas mediante los acuses de recibo.
- Indicar las coordenadas de la boyas en caso de que el nodo colector lo solicite.

### 5.1.1. Envío, enrutado y acuse de recibo de alarmas

Una vez que el sistema detecta un determinado evento acústico genera la alarma correspondiente y la añade, como una nueva línea, en un fichero que contiene todas las alarmas. Este archivo es constantemente chequeado por el software implementado en este módulo, a la espera de detectar nuevas entradas. Cuando se detecta una nueva alarma se mira la tabla de rutas por si existe algún camino válido hacia el nodo colector y, si es así, se envía directamente. En caso contrario, se inicia el proceso de Descubrimiento de Ruta. Si éste finaliza satisfactoriamente existirá, por lo tanto, una ruta válida bidireccional entre ambos nodos, y se enviará la alarma. Si no encuentra un camino entre ellos, esperará un determinado tiempo, fijado en la configuración del sistema, y volverá a comenzar el Descubrimiento de Ruta.

Cuando se envía una alarma no se manda la siguiente hasta que no se recibe el correspondiente acuse de recibo de la primera. Si, tras un determinado tiempo también fijado en la configuración, no hay noticia del nodo colector se volverá a enviar la misma alarma nuevamente. Este proceso se repetirá hasta que se reciba el correspondiente acuse de recibo o, en su defecto, se agote el número de reintentos determinados en la configuración del sistema. En este último caso, se descartará dicha alarma, ya que tras los reintentos y esperas dejará de ser útil. Únicamente, en caso de existir nuevas alarmas esperando, se iniciará el proceso de Descubrimiento de Ruta de nuevo.

Al trabajar todos los nodos con un protocolo de encaminamiento de redes ad hoc, cada uno de ellos tiene capacidad de enrutamiento y, por lo tanto, reenviará las alarmas de otros nodos que pasen por él. Por último, destacar que el nodo colector genera un registro de todas las alarmas que recibe durante la ejecución. De esta manera, es posible disponer de dicha información para su estudio más adelante.

### 5.1.2. Localización de nodos

Cada nodo de la red irá implementado en una embarcación, estructura flotante o, la mayor parte de las veces, en boyas. En muchas aplicaciones estas no irán ancladas al fondo marino sino que se podrán desplegar y recoger desde pequeñas embarcaciones y, durante el periodo de operación, estarán flotando libres en el mar. Por esta razón, se ha previsto que cada nodo de la red incluya un sistema de GPS (*Global Positioning System*) que pueda obtener en cada momento sus coordenadas geográficas exactas.

El nodo colector tiene la posibilidad de enviar una solicitud de posición en modo broadcast que llegue a todos los nodos de la red para que cada uno responda mediante un mensaje con sus coordenadas y la fecha y hora exacta de la consulta. Estos mensajes, inicialmente, se mandan únicamente bajo demanda pero, dependiendo de las aplicaciones finales que pueda tener el sistema, se podrá modificar para que se envíen periódicamente o según interese.

De esta manera, desde el nodo colector se puede realizar un registro de las posiciones de los demás nodos de la red. Esta información servirá de apoyo al estudio realizado con el registro de alarmas y, por supuesto, en muchos casos será vital para la recogida de las boyas al finalizar su utilización en el mar.

## 5.2. Protocolos de encaminamiento

Tal y como se refirió en la Sección 2.2, las redes ad hoc ofrecen grandes posibilidades, sobre todo en aquellos medios donde no existe una infraestructura de red preexistente. El diseño de protocolos de encaminamiento para este tipo de redes es actualmente un gran desafío debido a su naturaleza cambiante: tamaño, densidad de tráfico, rutas entre nodos, etc.

Existen una serie de criterios importantes a tener en cuenta en la selección o implementación de un protocolo de encaminamiento para una red ad hoc (Domingo-Aladrén, 2005):

- **Señalización mínima:**

La reducción del número de mensajes de control y del tamaño de estos ayuda a conservar la batería y el porcentaje de ciclo de trabajo disponible para transmitir, y a reducir la contienda de acceso al medio y, por lo tanto, a mejorar la comunicación entre los nodos.

- **Tiempo de procesamiento mínimo:**

Cuanto más simples sean las operaciones computacionales realizadas por el protocolo menor será el tiempo de procesamiento necesario y, por tanto, las baterías tendrán mayor tiempo de vida.

- **Mantenimiento en condiciones de topología dinámica:**

El protocolo debe ser capaz de detectar rupturas en los enlaces y detectar nuevas rutas alternativas lo antes posible.

- **Modo de operación distribuido:**

Propiedad esencial de las redes ad hoc.

- **Libre de bucles:**

Se debe evitar la existencia de paquetes perdidos circulando por la red, así como el envío de mensajes repetidos.

Existe una gran variedad de protocolos de encaminamiento para redes ad hoc. Con el objetivo de realizar la elección del más apropiado para su empleo en el módulo de entrega remota de datos del sistema LIDO DCL, en la siguiente subsección se realiza una posible clasificación de ellos atendiendo a su modo de creación de rutas bajo demanda o no.

### 5.2.1. Protocolos de encaminamiento proactivos, reactivos e híbridos

Una de las clasificaciones de protocolos de encaminamiento más extendidas es la siguiente (Royer y Toh, 1999):

- Protocolos proactivos, globales o basados en tablas
- Protocolos reactivos o bajo demanda
- Protocolos híbridos

A continuación, se presentan las rasgos principales de cada una de las categorías, para al final de la subsección analizar cual de ellas se ajusta más a las características deseadas.

### 5.2.1.1. Protocolos de encaminamiento proactivo

Mantienen información de encaminamiento actualizada entre todos los nodos de la red. Esta información se guarda en las tablas de encaminamiento, que periódicamente se actualizan para adaptarse a los cambios en la topología de la red.

Los protocolos proactivos se diferencian unos de otros según los criterios y el procedimiento de actualización de la información, el número de tablas de encaminamiento y las entradas de dichas tablas. Los protocolos proactivos más destacados son: OLSR (*Optimized Link State Routing*), basado en el algoritmo de estado de enlace, y DSDV (*Destination Sequenced Distance Vector*), basado en el algoritmo de vector distancia.

### 5.2.1.2. Protocolos de encaminamiento reactivo

Se obtiene la información de encaminamiento entre dos nodos bajo demanda, es decir, únicamente cuando existe una información que debe ser transmitida entre ellos. De este modo recursos como la energía o el ancho de banda se utilizan de una manera más eficiente que en el caso anterior. El inconveniente está en que el retardo por descubrimiento de ruta es considerablemente mayor.

El descubrimiento de rutas se realiza mediante el intercambio de mensajes de solicitud de ruta, RREQ (*Route Request*), y respuesta de ruta, RREP (*Route Reply*).

A su vez, los protocolos de encaminamiento reactivos pueden subdividirse en dos grupos:

- Basados en la fuente:

Los paquetes de datos transportan en su cabecera toda la ruta completa, de modo que los nodos intermedios deben consultar dicha cabecera para saber dónde deben reenviarlo. El protocolo más destacado de este grupo es el DSR (*Dynamic Source Routing*).

- Salto a salto:

Los paquetes llevan en su cabecera únicamente la dirección de destino y la del próximo nodo en la ruta. Cada uno se encarga de ir actualizando dicha cabecera según sus tablas de encaminamiento. Es necesario que cada nodo vaya actuando permanentemente su tabla de encaminamiento mediante el intercambio periódico de mensajes. El protocolo más destacado de este grupo es el AODV (*Ad hoc On-demand Distance Vector*).

### 5.2.1.3. Protocolos de encaminamiento híbridos

Básicamente, aprovechan las ventajas de los protocolos proactivos y reactivos. El procedimiento suele consistir en dividir la red en zonas, o bien árboles o clusters, para trabajar con diferentes tipos de nodos. De esta manera, cuando se buscan rutas entre nodos cercanos se utiliza encaminamiento proactivo y, por el contrario, entre nodos lejanos encaminamiento bajo demanda. Es común encontrar protocolos que utilizan información auxiliar, como posiciones mediante un GPS, para complementar estos protocolos. El más destacado de este grupo es el ZRP (*Zone Routing Protocol*).

### 5.2.1.4. Elección de un protocolo de encaminamiento

En esta sección se realiza un análisis cualitativo de las diferentes categorías de protocolos de enrutamiento presentadas con el objetivo de seleccionar la más adecuada para su utilización en el sistema LIDO DCL. Para otras aplicaciones, incluso en el medio marino, el análisis podría tener conclusiones distintas (Kong et al., 2008).

Por sus características, los protocolos de encaminamiento proactivo son especialmente adecuados en aquellas redes en que es necesario que el procedimiento de descubrimiento de ruta no tenga una latencia excesiva. Además, estas redes deben disponer de los recursos suficientes, como energía y ancho de banda, para asumir su funcionamiento. En el caso de LIDO DCL, la entrega inmediata de las alarmas no es un factor especialmente crítico. Este sistema se utilizará para formar perímetros de seguridad entorno a fuentes de ruido antropogénico en el medio marino. Así, existirá un margen de tiempo suficiente para poder entregar las alarmas y detener la emisión de ruido antes de que éste tenga efectos nocivos en ciertas especies. Por otro lado, como se vió en las Secciones 3.1 y 3.2, la energía y el ancho de banda si que son recursos limitados en el sistema. De esta manera se puede concluir que los protocolos de encaminamiento proactivo no son la opción más adecuada.

Por otra parte, los híbridos están especialmente pensados para redes extensas divididas en zonas. En ellos, se utilizan protocolos de encaminamiento proactivo para la comunicación entre nodos cercanos, y reactivo entre nodos lejanos. Inicialmente, el sistema LIDO DCL no está pensado para grandes poblaciones de nodos, por lo que la utilización de este tipo de protocolos tendría un resultado similar a la categoría anterior. Por ello, los protocolos de encaminamiento híbridos tampoco se consideran una opción apropiada.

Finalmente, están los protocolos reactivos entre los que diferenciamos dos tipos: basados en la fuente, como DSR, y salta a salto, como AODV. Anteriormente se comentó que la principal diferencia entre ellos es que el primer tipo transporta en la cabecera de sus paquetes la información de toda la ruta completa, mientras que el segundo únicamente lleva la dirección de destino y la del próximo nodo. Existen numerosos estudios comparando exclusivamente los protocolos de encaminamiento AODV y DSR (Das et al., 2000) (Domingo-Aladrén, 2005). Para evaluar que tipo es más apropiado, se destacan las siguientes características del sistema LIDO DCL:

- Limitación del ciclo de trabajo máximo en la banda 868 MHz: 10 %
- Tasa maxima de transmisión radio del módulo XBee-PRO 868: 24 kbps
- Generalmente no existe un gran flujo de alarmas.

Atendiendo a las dos primeras limitaciones el AODV supera al DSR, ya que tanto sus mensajes RREQ como los RREP utilizan una señalización menor. Además, de la tercera característica nombrada se extrae que es probable que entre el envío de diferentes alarmas la topología de la red vaya variando. En este caso, ambos protocolos deberán iniciar un nuevo proceso de búsqueda de ruta para cada alarma, siendo el del AODV el más óptimo de los dos. Cabe destacar que en el protocolo DSR cada nodo intermedio a lo largo de una ruta puede aprovechar la opción *promiscuous listening* y el hecho de que el encaminamiento esté basado en la fuente para aprender más rutas. Sin embargo, ya que en el sistema LIDO DCL existirá un nodo colector encargado de recoger todas las alarmas, las rutas se crearan hacia él y, por lo tanto, en este caso el protocolo AODV tendrá la misma ventaja.

En conclusión, los protocolos de encaminamiento reactivo salto a salto son la opción más adecuada para el módulo de entrega remota de datos. Dentro de este grupo, el más destacado, y del que se hará un estudio más en profundidad, es el AODV. Además, como veremos en las siguientes secciones, existen numerosos protocolos basados en él que proponen diferentes modificaciones para lograr una versión final más simple y óptima computacionalmente hablando.

Como se indicaba al inicio de esta subsección, el análisis realizado es cualitativo, ya que para un primer prototipo se considera que esto es suficiente. En versiones futuras, sería interesante y apropiado realizar pruebas con distintos protocolos de encaminamiento para obtener unos resultados cuantitativos y poder realizar una elección más precisa.

### 5.2.2. Protocolo AODV

#### 5.2.2.1. Visión general

AODV (Perkins et al., 2003) (Perkins y Royer, 1999) es un protocolo de encaminamiento reactivo salto a salto. Por ello, las rutas de cada nodo a cada nodo de la red no están permanentemente actualizadas, sino que se descubren y mantienen únicamente cuando es necesario. Sus principales características son:

- Señalización de control baja. Dado su carácter reactivo.
- Señalización de procesamiento mínima. Mensajes sencillos que requieren poco cálculo.
- Ausencia de bucles. Existen mecanismos específicos para su prevención.
- Funciona sólo con enlaces bidireccionales.

Con el fin de evitar bucles, cada nodo dispone de un número de secuencia (también llamado número de secuencia de destino) que permite mantener sólo la información de encaminamiento más reciente. Este número aumenta en una unidad cada vez que se envía un mensaje de solicitud de ruta, RREQ. Cuando un nodo recibe un RREQ del que es destinatario final, antes de responder con un mensaje de respuesta de ruta, RREP, debe actualizar su número de secuencia  $NumSec_D$  al valor máximo según la expresión:

$$NumSec_D = \text{Max}(NumSec_{D.\text{actual}}, RREQ.NumSec + 1) \quad (5.1)$$

donde  $NumSec_{D.\text{actual}}$  es su número de secuencia actual y  $RREQ.NumSec$  el número de secuencia del destino que se haya contenido en el RREQ.

En el caso de que un nodo reciba dos paquetes con el mismo identificador del nodo de origen pero diferentes números de secuencia, la información más reciente será la contenida en el paquete con el mayor de los dos. Si los números de secuencia son iguales se utilizará otra métrica para decidir, generalmente se obtará por la ruta con el menor número de saltos hacia el destino.

### 5.2.2.2. Declaración de aplicabilidad

El protocolo de encaminamiento AODV está especialmente diseñado para redes ad hoc dinámicas con poblaciones de entre decenas y miles de nodos móviles. AODV puede operar con tasas de movilidad bajas, medias y elevadas, así como con varios niveles de tráfico de datos. Está pensado para su utilización en redes donde todos los nodos pueden confiar en los demás, ya sea por el uso de claves preconfiguradas, o porque se sabe que no hay nodos intrusos maliciosos. AODV se ha diseñado para reducir la diseminación del tráfico de control, mediante la utilización del campo TTL (*Time To Live*, Tiempo de Vida), y eliminar la sobrecarga de cabeceras en el tráfico de datos, con el fin de mejorar la escalabilidad y el rendimiento.

### 5.2.2.3. Tabla de rutas y tabla de RREQ's

AODV utiliza tablas de rutas para almacenar la información de encaminamiento. Cada nodo de la red mantiene la suya propia, que tiene tantas entradas como destinos conoce. Cada entrada de la tabla cuenta con los siguientes campos:

- Dirección de destino
- Dirección del próximo salto
- Número de secuencia de destino
- Tiempo de vida (tiempo requerido para eliminar la ruta)

- Contador de saltos (número de saltos para alcanzar el destino)
- Otros indicadores de estado y rutas: validez del número de secuencia de destino, validez de ruta, si es o no reparable; y si lo es, si se está o no reparando.

Cada entrada de la tabla de rutas está asociada a un temporizador, de modo que, si una ruta no se utiliza durante un determinado periodo de tiempo expira y deja de ser válida. Sin embargo, cada vez que se emplea la información de la entrada o se recibe un mensaje de HELLO, el temporizador se actualiza.

Para cada entrada válida en la tabla de rutas de un determinado nodo, se mantiene una lista de precursores que contiene los nodos que utilizan dicha ruta como próximo salto en el camino hacia un nodo destino, es decir, los nodos vecinos que han recibido algún mensaje desde el nodo destino. Estos precursores recibirán notificaciones del nodo en caso de que se detecte algún problema en la ruta, como se explica más adelante en el proceso de Mantenimiento de Ruta.

Además, cada nodo mantiene un registro de los mensajes RREQ que recibe. Utiliza los campos RREQ ID y dirección de origen, que identifican inequívocamente cada RREQ para poder descartar los repetidos y optimizar el rendimiento general de la red.

#### 5.2.2.4. Formato de mensajes

El protocolo de encaminamiento AODV define los siguientes tipos de mensajes:

- **Mensaje RREQ**

El mensaje RREQ se utiliza para descubrir una ruta hacia un determinado nodo destino. En la Figura 5.1 se muestra el formato del mismo, que incluye los siguientes campos:

- Tipo: indica el tipo de mensaje.
- J: Join flag; utilizado cuando el nodo de origen quiere formar parte de un grupo multicast.
- R: Repair flag; utilizado cuando un nodo se dispone a iniciar una reparación de dos partes de un árbol multicast que se han desconectado.
- G: Gratuitous RREP flag; indica si es necesario enviar en modo unicast un RREP Gratuito al nodo indicado en el campo *dirección de destino*.
- D: Destination only flag; indica que únicamente el nodo destino puede responder a este mensaje RREQ.
- U: Unknow sequence number flag; indica que el campo *número de secuencia de destino* es desconocido.
- Reservado: Bits reservados. Se envían con el valor 0, y son ignorados en la recepción.
- Contador de saltos: indica el número de saltos desde el nodo de origen hasta el nodo que está procesando este mensaje RREQ.
- RREQ ID: número de secuencia que junto con la dirección del nodo origen identifica inequívocamente el mensaje RREQ. Se incrementa cada vez que un nodo inicia el envío de uno de ellos.
- Dirección de destino: identificador del nodo con el que se quiere realizar la comunicación.

- Número de secuencia de destino: el último número de secuencia recibido en el nodo origen de cualquier ruta hacia el nodo destino.
- Dirección de origen: identificador del nodo que inicia el descubrimiento de ruta.
- Número de secuencia de origen: número de secuencia actual del nodo origen.

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1	2 3 4 5 6 7 8 9 0 1	2 3 4 5 6 7 8 9 0 1	3
Tipo	J R G D U	Reservado	Contador de saltos
RREQ ID			
Dirección de destino			
Número de secuencia de destino			
Dirección de origen			
Número de secuencia de origen			

Figura 5.1: Formato del mensaje RREQ del protocolo AODV.

#### ■ Mensaje RREP

El mensaje RREP es la contestación a un RREQ. Indica que existe una ruta hacia el nodo destino. En la Figura 5.2 se muestra el formato del mismo, que incluye los siguientes campos:

- Tipo: indica el tipo de mensaje.
- R: Repair flag; utilizado cuando un nodo se dispone a iniciar una reparación de dos partes de un árbol multicast que se han desconectado.
- A: Acknowledgment required flag; se utiliza para comprobar si un enlace es unidireccional.
- Reservado: Bits reservados. Enviados con el valor 0, y son ignorados en la recepción.
- Longitud del prefijo: si no es cero, especifica que el siguiente salto indicado puede ser utilizado por nodos con el mismo prefijo, como si fuesen los destinatarios de la petición.
- Contador de saltos: indica el número de saltos desde el nodo de origen hasta el nodo que está procesando este mensaje RREQ.
- Dirección de destino: identificador del nodo con el que se quiere realizar la comunicación.
- Número de secuencia de destino: el último número de secuencia recibido en el nodo origen de cualquier ruta hacia el nodo destino.
- Dirección de origen: identificador del nodo que inicia el descubrimiento de ruta.
- Tiempo de vida: tiempo en milisegundos durante el cual los nodos que han recibido el mensaje RREP deben considerar válida la ruta.

0	1	2	3	
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1	
Tipo	R   A	Reservado	Long. Pref.	Contador de saltos
Dirección de destino				
Número de secuencia de destino				
Dirección de origen				
Tiempo de vida				

Figura 5.2: Formato del mensaje RREP del protocolo AODV.

#### ■ Mensaje RERR (*Route Error*)

El mensaje RERR se utiliza si se rompe algún enlace provocando que uno o más nodos se vuelvan inalcanzables desde ciertos nodos vecinos. En la Figura 5.3 se muestra el formato del mismo, que incluye los siguientes campos:

- Tipo: indica el tipo de mensaje.
- N: No delete flag; se utiliza cuando se está realizando una reparación local (*local repair*) de un enlace roto y el nodo receptor del mensaje RERR no debe borrar ninguna ruta.
- Reservado: Bits reservados. Enviados con el valor 0, y son ignorados en la recepción.
- Contador de destinos: el número de destinos inalcanzables que se incluyen en el mensaje. El valor mínimo de este campo es uno.
- Dirección de destino inalcanzable: identificador del nodo destino que actualmente es inalcanzable debido a la rotura del enlace.
- Número de secuencia de destino inalcanzable: el número de secuencia del nodo destino inalcanzable indicado en el campo anterior que se encuentra en la entrada de la tabla de rutas.

0	1	2	3	
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1	
Tipo	N	Reservado	Contador destinos	
Dirección de destino inalcanzable				
Número de secuencia de destino inalcanzable				
Dirección de destino inalcanzable adicional (si es necesario)				
Número de secuencia de destino inalcanzable (si es necesario)				

Figura 5.3: Formato del mensaje RERR del protocolo AODV.

#### 5.2.2.5. Descubrimiento de Ruta

Cuando un nodo de la red necesita comunicarse con otro, en primer lugar, debe comprobar si en su tabla de rutas existe algún camino válido hacia el nodo destino. De ser así, el nodo podría comenzar a enviar su información sin necesidad de realizar ninguna otra operación. En el caso contrario, comenzará el proceso de Descubrimiento

de Ruta, que si concluye satisfactoriamente proporcionará la información necesaria al nodo origen para poder realizar la comunicación (Véase la Figura 5.4).

En el proceso pueden distinguirse dos fases: la formación del camino de vuelta y la formación del camino de ida. La primera de ellas establece todas las rutas posibles entre el origen y el destino, trazadas por el recorrido de los mensajes RREQ que se transmiten en modo difusión o *broadcast*. La segunda determina el itinerario que finalmente seguirán los paquetes desde un nodo al otro.

A continuación, se describen ambas fases. El material ilustrativo creado para ello está basado en el documento (Klein-Berndt, 2001) y en unas imágenes creadas por el profesor Nitin Vaidya incluidas en (Domingo-Aladrén, 2005).

#### ■ Formación del camino de vuelta

Como representa la Figura 5.4b, el nodo origen, al no encontrar una ruta válida hacia el nodo destino en su tabla de rutas, genera un mensaje RREQ y lo envía en modo broadcast. Este mensaje podrá ser respondido con un RREP por el nodo destino o por cualquier nodo de la red que conozca una ruta actualizada hacia él.

Cuando un nodo intermedio obtiene un RREQ debe comprobar si ya ha recibido anteriormente dicho mensaje. Para ello, mira los campos RREQ ID y dirección de origen y verifica si dicha información se encuentra en alguna entrada de su tabla de registro de RREQ's. Si es así, el nodo descarta el paquete, como ocurre por ejemplo en la Figura 5.4d donde el nodo C recibe los RREQ de los nodos G y H, pero no hace una retransmisión porque ya la hizo una vez anteriormente, concretamente en la Figura 5.4c. Si no ha recibido anteriormente dicho mensaje, el nodo registra la recepción del RREQ y crea una entrada en su tabla de rutas que contiene el camino hacia el nodo origen, es decir, el de vuelta (Véase la Figura 5.4c). Esta ruta tiene un tiempo de vida determinado tras el que es eliminada. El camino de vuelta tiene su utilidad cuando el nodo recibe más tarde un RREP que debe ser entregado al nodo origen a través del mismo.

Para poder responder a un mensaje RREQ, un nodo debe ser el destinatario final del mismo o verificar que:

- Exista en la tabla de rutas una entrada con un camino hacia el destino que no haya expirado.
- El número de secuencia de destino de dicha entrada en la tabla de rutas sea mayor o igual que el número de secuencia de destino del mensaje RREQ, es decir:

$$NumSec_{tabla} \geq NumSec_{RREQ} \quad (5.2)$$

Si esto se cumple, el nodo incrementa el contador de número de saltos del RREQ y genera el correspondiente mensaje RREP. Si no, el nodo incrementa el contador del número de saltos del RREQ y reenvía dicho mensaje en modo broadcast para continuar con el Descubrimiento de Ruta.

Para evitar un consumo excesivo de recursos, el protocolo cuenta con un mecanismo conocido como búsqueda expansiva en anillo (*expanding ring search*). Éste consiste en inicialmente asignar a los mensajes RREQ un TTL pequeño, tras el cual el mensaje se descarta. Si en un determinado tiempo no se recibe un RREP, este valor se va incrementando progresivamente en los siguientes reintentos de Descubrimiento de Ruta, hasta alcanzar el número máximo de ellos.

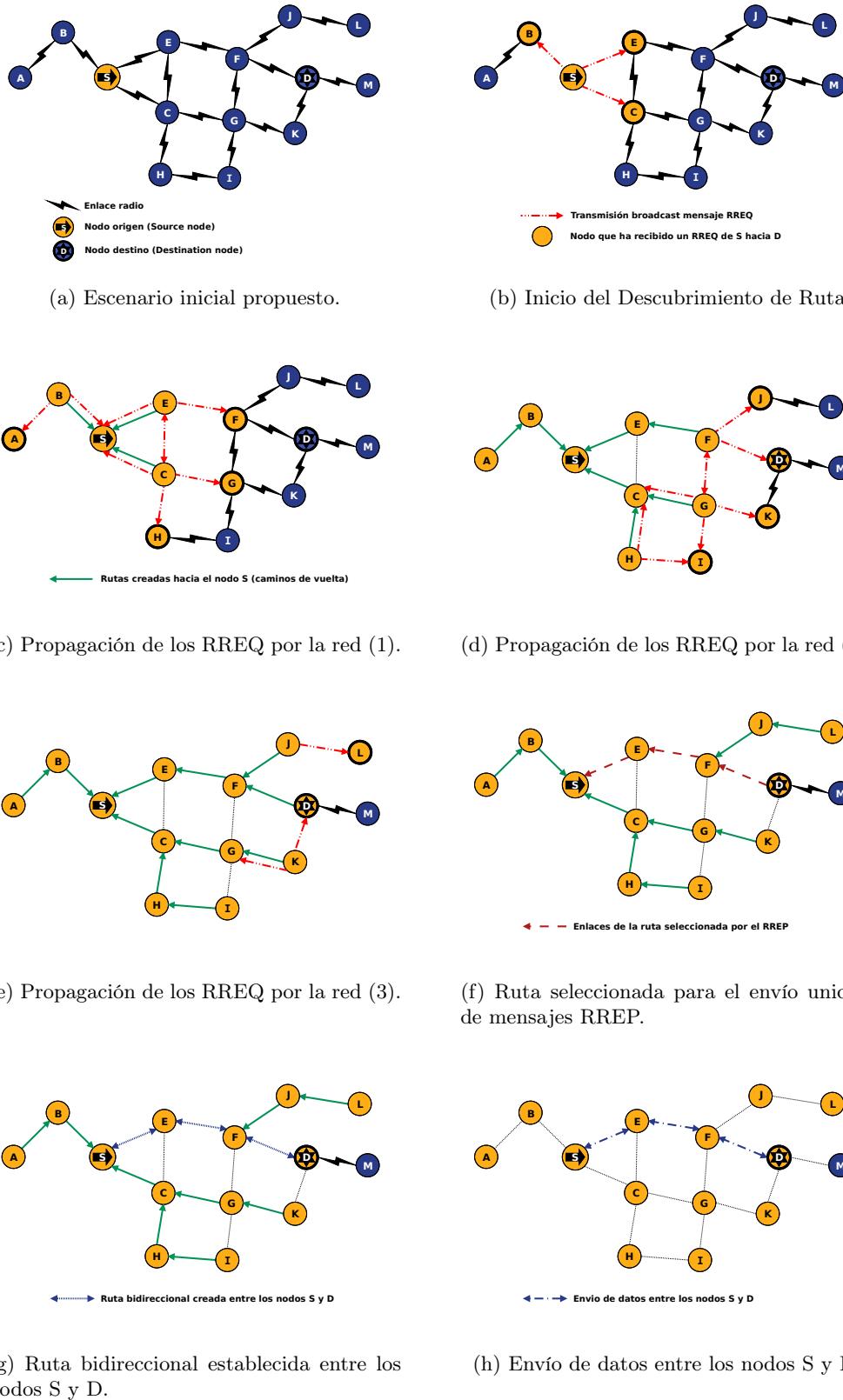


Figura 5.4: Proceso de Descubrimiento de Ruta en una red ad hoc que utiliza el protocolo de encaminamiento AODV para enviar datos desde un nodo origen S hasta un nodo destino D.

### ■ Formación del camino de ida

Dependiendo de si es el nodo destino el que responde el RREQ, o es un nodo intermedio, se procede de una u otra manera. Concretamente en el primer caso el nodo destino:

- Introduce su número de secuencia en el paquete de acuerdo con la expresión 5.1.
- Sitúa el contador de saltos a cero.

Si es un nodo intermedio el que inicia el envío del RREP, procede de la siguiente manera:

- Introduce el número de secuencia del nodo destino en el paquete. Este valor se encuentra en la correspondiente entrada de su tabla de rutas.
- Sitúa el contador de saltos al mismo valor que el número de los mismos que tiene en su ruta hacia el nodo destino.

A continuación, en ambos casos se asigna un valor determinado al temporizador de tiempo de vida y, por último, se envía en modo unicast el mensaje según el camino de vuelta formado en la primera fase del proceso de Descubrimiento de Ruta.

Cuando un nodo intermedio recibe un mensaje RREP incrementa el contador de saltos del paquete en una unidad y guarda en su tabla de rutas la información del camino de ida hacia el nodo destino, utilizando como próximo salto el nodo del cual recibió el RREP. Finalmente, reenvía este paquete al siguiente nodo en dirección a la fuente.

Podría producirse la situación de que un nodo reciba un RREP por parte de más de un nodo vecino. En tal caso, reenviaría el primero de ellos, y un segundo únicamente en caso de que el número de secuencia de destino contenido en el paquete fuese mayor, o el valor del contador de saltos menor, actualizando en tal caso su tabla de rutas. Este mismo criterio es utilizado por el nodo origen para establecer el camino de ida al recibir varios paquetes RREP. Una vez establecido este camino, es posible comenzar a utilizarlo para el envío bidireccional de paquetes de datos (Véase la Figura 5.4h).

En la Figura 5.4f vemos como el nodo destino tiene dos opciones, a través del nodo F o del K, para establecer el camino de vuelta. De ellas, opta por la que tiene el menor número de saltos, pero esto no siempre implica que se trate de la trayectoria más óptima de toda la red. En el protocolo de encaminamiento AODV los nodos únicamente aceptan y procesan un RREQ, rechazando aquellos paquetes, con el mismo RREQ ID y dirección de destino, que se reciben posteriormente. De este modo, algunas trayectorias más óptimas que la ruta final podrían no descubrirse al utilizar algún nodo común.

#### 5.2.2.6. Mantenimiento de Ruta

El protocolo de encaminamiento AODV está especialmente pensado para operar en el medio radio y con nodos móviles. Esto hace común la rotura de enlaces entre ellos. Además, como ya se ha visto, las rutas que se descubren tienen un tiempo de vida durante el que pueden ser utilizadas y que al terminar las inhabilita. Esto permite

que no sea necesario establecer un camino cada vez que se quiere mandar un mensaje de datos pero, a su vez, hace necesaria la existencia de algún proceso encargado de la monitorización del estado de los enlaces y de su gestión en caso de rotura. Este proceso se conoce como Mantenimiento de Ruta.

El protocolo AODV utiliza mensajes HELLO, que contienen la dirección del nodo, su número de secuencia actual y el tiempo de vida de los enlaces, y que cada nodo envía periódicamente en modo broadcast a sus vecinos para comprobar si existe conectividad entre ellos. De este modo, cada nodo puede actualizar la información referente a sus vecinos en su tabla de rutas. Si durante un determinado intervalo de tiempo un nodo deja de recibir estos mensajes de alguno de sus vecinos, elimina la entrada asociada a dicho vecino de su tabla de rutas. La utilización de mensajes HELLO no es necesaria si existe algún otro mecanismo que realice su tarea, como puede ser retroalimentación procedente de la capa de enlace de datos, LLN (*Link Layer Notification*).

Cuando un nodo detecta un fallo en un enlace, envía un mensaje de error de ruta, RERR (*Route Error*), hacia el nodo fuente para informar de los destinos que ya no pueden alcanzarse. Si existen varios nodos precursores (entre el nodo origen y el nodo que detectó la rotura) que estaban utilizando el enlace, el mensaje RERR se envia en modo broadcast, en caso contrario se envia en modo unicast.

Al recibir un paquete de RERR, un nodo verifica que efectivamente la rotura afecta a alguna de sus entradas en la tabla de rutas, la invalida y por último reenvía el mensaje hacia la fuente. Cuando es el nodo origen el que recibe el RERR, éste puede optar por iniciar un nuevo proceso de Descubrimiento de Ruta si lo considera necesario.

AODV también cuenta con algunas opciones de optimización, como la posibilidad de realizar reparaciones locales de los enlaces. Cuando un nodo detecta la rotura de un enlace, en vez de mandar inmediatamente un mensaje RERR a la fuente, envía un RREQ hacia el nodo destino con el número de secuencia de destino incrementado en una unidad. Mientras espera el RREP va almacenando los paquetes de datos. Si tras un determinado tiempo no se recibe respuesta, la reparación local no habrá tenido éxito y será necesario avisar a la fuente con un mensaje RERR.

Otra característica adicional del protocolo AODV consiste en enviar mensajes RREP gratuitos desde un nodo intermedio hacia el nodo destino para informarle de que el propio nodo intermedio ha respondido un RREQ. Esto permite que el nodo destino conozca una ruta hacia el nodo origen por si la comunicación fuese bidireccional. De otro modo, el nodo destino no tendría noción de que existe tal ruta hacia él y, por lo tanto, no podría responder a la fuente si tuviera que hacerlo.

### 5.2.3. Protocolos basados en el AODV

El protocolo de encaminamiento AODV es uno de los más estudiados para su utilización en redes ad hoc móviles, también conocidas como MANET (*Mobile Ad hoc Network*). Esto ha dado lugar a la aparición de numerosos protocolos que proponen diferentes modificaciones sobre el original, según la aplicación final que se les quiera dar. En la presente subsección se realiza una revisión de los principales protocolos, orientados a redes de sensores, que están basados en el AODV (Gomez et al., 2006).

#### 5.2.3.1. AODVjr

AODVjr (Chakeres y Klein-Berndt, 2002) (Klein-Berndt y Chakeres, 2002) es una de las primeras versiones simplificadas del AODV. Los autores muestran a través de

varias simulaciones unos resultados muy próximos a los obtenidos con el protocolo AODV completo. Al mismo tiempo, reducen significativamente la complejidad de implementación de éste al eliminar varias características definidas en su especificación original. En primer lugar, dejan de utilizarse los números de secuencia. El protocolo AODVjr evita la formación de bucles definiendo que únicamente el nodo destino puede generar los mensajes RREP. Esto también elimina la necesidad del envío de mensajes RREP gratuitos desde nodos intermedios. Además, este protocolo ya no emplea el número de saltos de las rutas como métrica. En su lugar establece que la ruta relacionada con el primer mensaje RREP que recibe el nodo origen es la elegida. Por último, se eliminan los mensajes HELLO, RERR y la lista de precursores, y el mantenimiento de ruta pasa a realizarse de la siguiente manera. Si la comunicación es unidireccional, el nodo destino manda mensajes de conexión (*connect messages*) al nodo origen. Si es bidireccional, no se utilizan mensajes adicionales; un nodo origen detecta la rotura de una ruta cuando deja de recibir mensajes desde el nodo destino.

#### 5.2.3.2. AODVbis

AODVbis es una revisión de la especificación original del AODV que aclara algunos aspectos de funcionalidad y deja varias características como opcionales. La utilización de mensajes RERR ya no es obligatoria. En caso de utilizarse, estos se envían en modo broadcast localmente, de modo que se evita tener que mantener una lista de precursores para cada entrada de la tabla de rutas. Cada nodo que reciba un RERR debe añadir los destinos inalcanzables que provienen de procesar dicho mensaje a los nuevos mensajes RERR que sean generados. El contador de saltos también deja de ser la métrica obligatoria para las rutas, y por defecto se procede como en el caso del AODVjr. También se eliminan las reparaciones locales y se añade la característica acumulación de rutas (*path accumulation*), que permite a un determinado nodo adquirir información de enrutamiento extra de una lista con el camino recorrido incluida en los mensajes RREQ y RREP que pasan por él.

#### 5.2.3.3. LoWPAN-AODV

LoWPAN-AODV es una propuesta para adaptar AODV a entornos LoWPAN (*Low-Rate Wireless Personal Area Network*), como es el caso de las redes de sensores. AODVbis es la especificación por defecto del LoWPAN-AODV, salvo por algunos aspectos que conserva de la filosofía del AODVjr. Los únicos mensajes de control definidos son RREQ y RREP. Sólo el nodo destino puede responder un RREQ con un RREP. La conectividad local se mantiene por el mecanismo LLN, según establece el estándar IEEE 802.15.4. En este caso, el contador de saltos sí que es la métrica utilizada para determinar las mejores rutas.

#### 5.2.3.4. LOAD

LOAD, al igual que AODVjr y LoWPAN-AODV, sólo permite que el nodo destino genere los mensajes RREP, evitando así el uso de números de secuencia. Permite las reparaciones locales, y en caso de que estas no finalicen satisfactoriamente, el nodo que detectó el fallo del enlace debe generar y enviar en modo unicast un mensaje RERR al nodo origen. La métrica de encaminamiento utilizada en LOAD es el coste acumulado del enlace entre el origen y el destino, usando el indicador de calidad de enlace (*Link Quality Indicator*).

*Quality Indicator (LQI)*) de la capa física del IEEE 802.15.4 como entrada para el cálculo. Finalmente, como en el LoWPAN-AODV, se utiliza el mecanismo LLN para evaluar la conectividad de las rutas.

#### 5.2.3.5. TinyAODV

TinyAODV es una implementación minimalista del AODV para dispositivos que utilizan el sistema operativo TinyOS. En este protocolo, si un paquete de datos debe enviarse y no existe una ruta válida hacia el destino, se inicia el proceso de Descubrimiento de Ruta, pero el paquete que requirió la ruta es descartado. De este modo, el siguiente paquete de datos será el primero en utilizar la ruta descubierta. El número de reintentos de Descubrimiento de Ruta se puede modificar, pero su valor por defecto es tres.

De igual modo que algunos de los protocolos anteriores, sólo el nodo destino puede generar los mensajes RREP. El mecanismo LLN puede utilizarse para detectar fallos en los enlaces, aunque por defecto no está habilitado. Por ello, TinyAODV está especialmente pensado para redes con topologías estáticas, donde no se esperan apenas fallos en los enlaces. Si un paquete de datos no puede entregarse por un fallo en un enlace, éste será descartado. A continuación, se generará un mensaje RERR que será enviado en modo broadcast localmente, como hacía el AODVbis. Por último, la reparación local no está soportada, y el contador de saltos es la métrica utilizada para determinar las mejores rutas.

#### 5.2.3.6. NST-AODV

Not So Tiny - AODV (NST-AODV) es una implementación de un protocolo de encaminamiento en nesC para plataformas TinyOS. Fué diseñado con el objetivo de añadir varias prestaciones adicionales a las básicas del TinyAODV con el objetivo de lograr un mejor rendimiento, especialmente en redes dinámicas.

Sus características principales son:

- El mecanismo LLN está activado por defecto, ya que se asume que el protocolo operará en redes con topología dinámica.
- El paquete de datos que dispara un proceso de Descubrimiento de Ruta no es descartado, y en cuanto la ruta está disponible es enviado.
- Tras una transmisión fallida a nivel de enlace, desde la capa de red se pueden realizar hasta dos reintentos adicionales.
- Si un paquete no puede ser enviado tras los tres intentos de la capa de red, será almacenado en un buffer y enviado si se encuentra una nueva ruta. Esta operación puede realizarla tanto el nodo origen como un nodo intermedio que repare localmente la ruta.
- Se han añadido dos colas FIFO (*First In, First Out*): una para guardar los paquetes recibidos mientras se realiza un Descubrimiento de Ruta, y otra para los paquetes a enviar. Ambas colas pueden dimensionarse según interese en su configuración.
- Un nodo intermedio puede generar un mensaje RREP si conoce una ruta válida al destino solicitado en el RREQ.

### 5.2.4. Definición del protocolo de encaminamiento AODV-LAB

Como se vió en la Sección 4.1, actualmente no existen en el mercado módulos de comunicación de gran alcance en la banda de 868 MHz que incorporen en su firmware el soporte para una topología de malla dinámica o ad hoc. Por ello, surgió la necesidad de incorporar esta funcionalidad a los módulos de comunicación seleccionados. En primer lugar, se realizó una búsqueda de protocolos de encaminamiento reactivos ya desarrollados que pudiesen ser válidos para el proyecto, pero únicamente se encontraron implementaciones orientadas a su utilización en: Internet, el simulador de redes ns-2 y el sistema operativo TinyOS. Por ello, finalmente se decidió desarrollar en C/Linux un protocolo de encaminamiento propio, llamado AODV-LAB, para poder utilizarlo con el módulo de entrega remota de datos, que en su versión final irá implementado en un PC embebido con el sistema operativo GNU/Linux.

A continuación, a partir del análisis de los protocolos presentados en la Subsección 5.2.3, se definen las características principales del AODV-LAB:

- Para evitar bucles y la utilización de mensajes RREP gratuitos, únicamente el nodo destino puede generar mensajes RREP. Además, se mantienen los números de secuencia como en el AODV. Esto permite diferenciar los mensajes RREQ y RREP más recientes. De este modo, en el AODV-LAB cuando un nodo recibe uno de estos mensajes, en primer lugar da prioridad a la información de encaminamiento más actual y, únicamente en caso de que el número de secuencia del mensaje coincida con el de la tabla de rutas, utilizará el contador de saltos para optar por el camino con menor número de ellos. Como mejora en un futuro, se contempla añadir sistemas de evaluación de la calidad de los enlaces como criterio para la selección de rutas que, especialmente en el medio marino, puede suponer importantes mejoras.
- Se eliminan los mensajes HELLO, RERR, y la lista de precursores, reduciendo la complejidad de implementación, la utilización de recursos del sistema y manteniendo unas características de funcionamiento similares, como se explicó anteriormente en el caso de AODVjr.
- Como mecanismo para el mantenimiento de la conectividad se utiliza el envío de mensajes de acuse de recibo de datos, llamados ACK (*Acknowledgement*). Si tras un periodo de tiempo definido un nodo que ha enviado un paquete de datos no recibe su correspondiente ACK, volverá a enviar el paquete tantas veces como se indique en el número de reintentos. Una vez agotados se dará la ruta por inválida y se volverá a iniciar el proceso de Descubrimiento de Ruta. El empleo de un número de reintentos por encima de la capa de enlace se utiliza para evitar que por pérdidas de conexión temporales y esporádicas, o por pequeños periodos con peores condiciones para la transmisión radio, se inicien nuevos Descubrimientos de Ruta innecesarios, con el gasto de recursos que esto supone.
- No se contempla la utilización del mecanismo LLN ya que éste depende del hardware utilizado, aunque en un futuro podría suponer una mejora considerable del sistema gracias a sus numerosas ventajas: detección más rápida de roturas de enlace, utilización más eficiente del ancho de banda y menor consumo de energía.
- Cuando una ruta se da como inválida, el paquete que agotó sus reintentos es descartado. Esto es así ya que en ese momento habrá transcurrido demasiado

tiempo y dicha alarma ya no será útil, por lo que enviarla únicamente supondría un gasto de recursos.

- Existe una cola FIFO (*First In, First Out*) para los mensajes de entrada, de modo que los mensajes serán procesados en el orden en que se recibieron.
- No se realiza reparación local de rutas. La aplicación principal de este sistema será crear perímetros de seguridad alrededor de fuentes de contaminación acústica. Estos perímetros no requerirán a priori rutas de más de 3 saltos entre un nodo origen y uno destino, por lo que la reparación local en este caso no tiene mucho sentido, ya que este mecanismo está especialmente pensado para grandes redes.

### 5.2.5. Comparación entre los distintos protocolos de encaminamiento

En la Tabla 5.1 se muestra un resumen de todos los protocolos de encaminamiento basados en el AODV que se han presentado anteriormente.

Nombre del protocolo	Mecanismo de mantenimiento de conectividad	Mensaje RERR	Lista precursores	Reparación local	Sólo el nodo destino genera RREP	Métrica de encaminamiento	Especificación y/o estado de implementación
AODV	Mensajes HELLO, LLN, etc.	Si	Si	Si (puede)	No	Contador de saltos	Implementado para múltiples plataformas
AODVjr	Mensajes de conexión (tráfico unidir.)	No	No	No	Si	Primer RREP recibido	Implementado para el simulador ns-2
AODVbis	Mensajes HELLO, LLN, etc.	Si (debe)	No	No	No	Primer RREP recibido	Proyecto Internet expirado
LowPAN-AODV	LLN	No	No	No	Si	Contador de saltos	Proyecto Internet expirado
LOAD	LLN	Si (puede)	No	Si (puede)	Si (debe)	Basado en LQI	Proyecto Internet
TinyAODV	LLN (deshabilitado por defecto)	Si	No	No	Si	Contador de saltos	Implementado para TinyOS
NST-AODV	LLN	Si	No	Si	No	Contador de saltos	Implementado para TinyOS
AODV-LAB	Mensajes ACK y timeouts	No	No	No	Si	Contador de saltos	Implementado en C/Linux

Tabla 5.1: Comparación de protocolos de encaminamiento basados en el AODV

## 5.3. Implementación en C/Linux

### 5.3.1. Introducción y justificación

El sistema LIDO DCL, en su primera aproximación, introducirá muy poco tráfico en la red. Las alarmas correspondientes a las distintas vocalizaciones de cetáceos no son muy frecuentes ya que la presencia de ellos no es continua y, además, el sistema no enviará una por sonido detectado sino que tratará de identificar la presencia de individuos en una determinada zona y avisar de esta periódicamente mientras continúen allí. Esto ha permitido, en la realización de este proyecto, centrarse directamente en la implementación de un prototipo sin la necesidad previa de: llevar a cabo un estudio

exhaustivo en un simulador de redes del protocolo de encaminamiento, realizar comparaciones con otros ya existentes y, finalmente, buscar la mejor solución. El objetivo es dar una primera aproximación al problema y demostrar su viabilidad en condiciones reales.

Se destaca la importancia de realizar una implementación de un protocolo de encaminamiento (Chakeres y Belding-Royer, 2004). La simulación es una gran herramienta para el desarrollo de los mismos, pero esta no garantiza que un protocolo funcione en la práctica. Los simuladores realizan suposiciones y utilizan modelos simplificados que pueden no reflejar del todo el funcionamiento real de la red. Por otro lado, en un simulador el código se encuentra contenido en una única unidad lógica, que está claramente definida. Sin embargo, una implementación requiere la utilización de un sistema con muchos componentes, algunos incluso con poca documentación o sin ella. Esto obliga al desarrollador a comprender no solo el protocolo de encaminamiento, sino también todos los componentes del sistema y sus interacciones.

Como ya se comentó en la Subsección 5.2.4, para este proyecto en primer lugar se trató de encontrar alguna implementación ya desarrollada para utilizarla al completo o como punto de partida. De esta búsqueda resultaron principalmente dos grupos que, como se explica a continuación, no se ajustan del todo a lo requerido:

- Implementaciones orientadas a Internet (Chakeres y Belding-Royer, 2004) (Kawadia et al., 2003). Son muy complejas. Van integradas en el kernel de Linux, utilizando para su aplicación muchas señales e indicadores que para este proyecto no son necesarias y lo complican considerablemente.
- Implementaciones para el sistema operativo TinyOS (Gomez et al., 2006) (Wang et al., 2006). TinyOS es un sistema operativo de código abierto diseñado para dispositivos inalámbricos de bajo consumo como los utilizados en redes de sensores. El sistema LIDO DCL posee algunas características de las redes de sensores pero, al trabajar con audio y tener que procesarlo en tiempo real, requiere de un sistema operativo más potente, lo que a su vez supone un mayor consumo energético. Concretamente operará sobre GNU/Linux.

Por último, tal y como se puede ver en la Subsección 4.2.1, los módulos XBee poseen un modo de operación transparente en el que se comportan como una conexión serie. Esto nos permite crear tramas propias, enviarlas al módulo como si se hiciera hacia el puerto serie y recibirlas del mismo modo.

Por todo ello, como muestra la Figura 5.5, finalmente se ha optado por realizar una implementación del sistema como un programa en C/Linux donde el módulo de entrega remota de datos utiliza el protocolo de encaminamiento AODV-LAB para encontrar rutas hacia el nodo colector y poder mandar sus alarmas y recibir los correspondientes ACKs. A continuación se detallará la organización del código, las tramas que utiliza y el flujo del programa.

### 5.3.2. Organización del código

En la Figura 5.6 se muestra como está organizado el código implementado para este proyecto. Dentro de cada fichero se han indicado los archivos con extensión .c y las funciones que contiene cada uno. Para darle una mayor claridad al diagrama se han omitido los archivos con extensión .h que contienen los prototipos de las funciones y

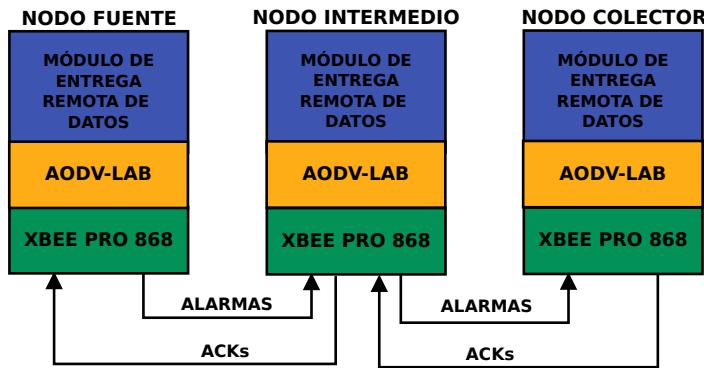


Figura 5.5: Arquitectura de operación del módulo de entrega remota de datos.

definiciones de variables. Las cabeceras de las funciones más importantes se pueden consultar en el Apéndice C.

### 5.3.3. Tramas

En el sistema únicamente se definen seis tipos de tramas. En la Tabla 5.2 se indican sus nombres, identificadores y una breve descripción de cada una. Posteriormente, se detallan todas ellas indicando, en cada caso, los campos de la trama y una especificación de los mismos.

Nombre de la trama	Identificador	Descripción
RREQ	Q	Mensaje de solicitud de ruta
RREP	P	Respuesta a solicitud de ruta
DATA	D	Mensaje de datos de alarma
ACK	A	Acuse de recibo de datos
PREQ	W	Mensaje de solicitud de posición (Where are you?)
PREP	C	Mensaje de posición (Coordinates)

Tabla 5.2: Tramas definidas en el sistema

En el programa en C/Linux se utilizan corchetes para delimitar cada trama. De esta manera se evita procesar mensajes incompletos. Además, los campos van separados por el carácter "|". Esto permite no asignarles un tamaño determinado. Así, por ejemplo, la dimensión de una trama que contenga el campo *número de secuencia* irá aumentando según aumente el número de dígitos de éste, y no utilizará desde el principio un tamaño definido. Esto ayuda a reducir la cantidad de información transmitida. En este proyecto no se ha trabajado la compresión de las tramas pero, como futuro trabajo, supondría una mejora para el sistema. Todos los campos se han definido de manera parametrizable dotando al código de una mayor versatilidad. Por último, indicar que el orden de

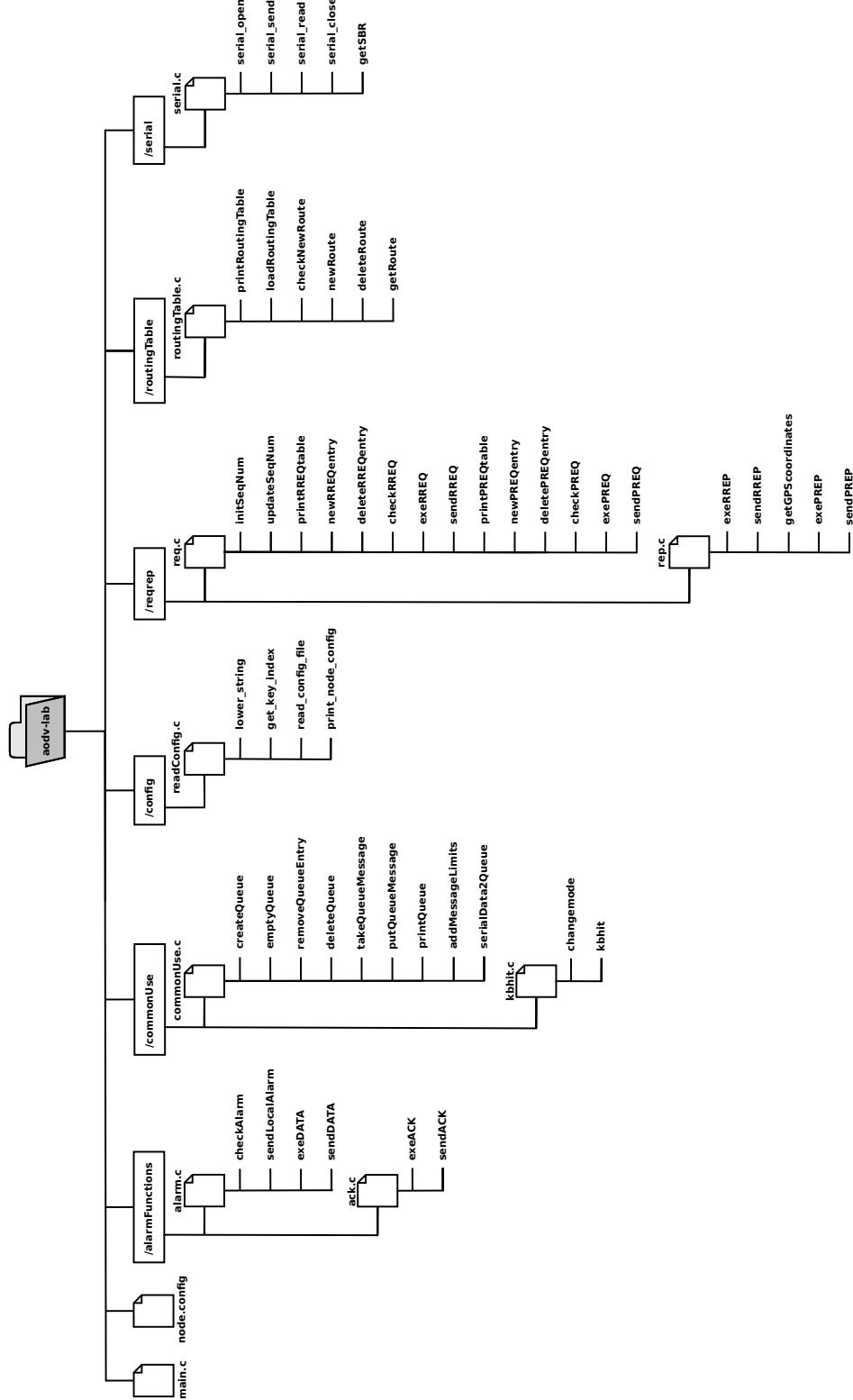


Figura 5.6: Organización del código implementado.

los campos en las tramas está determinado por la necesidad de la información que contienen en el procesado de los mismos por el sistema.

### 5.3.3.1. Trama RREQ

Se describe en la Tabla 5.3.

Campos de la trama RREQ	Identificador	Definición	Descripción
Next Node ID	NNID	<code>char nnid[NID_BYTES];</code>	Indica el destinatario inmediato de la trama, es decir, el próximo nodo en la ruta hacia el destinatario final. El valor 0 se corresponde con el modo Broadcast.
Frame ID	frameID	<code>char frameID[FID_BYTES];</code>	Indica el tipo de trama.
Source Node ID	SNID	<code>char snid[NID_BYTES];</code>	Indica el nodo que inició la búsqueda de ruta hacia el nodo destino indicado.
Sequence Number	seqN	<code>unsigned long int seqN;</code>	Número de secuencia. Se utiliza para evitar bucles e identificar la información más reciente.
Destination Node ID	DNID	<code>char dnid[NID_BYTES];</code>	Indica el nodo destino por el que se inició la búsqueda. En esta aplicación coincidirá con el nodo colector.
Previous Node ID	PNID	<code>char pnid[NID_BYTES];</code>	Indica el predecesor inmediato en la ruta, es decir, al nodo del que hemos recibido la trama. El nodo fuente se pone a sí mismo como predecesor.
Hop Counter	HOPS	<code>int hops;</code>	Indica el número de saltos en la ruta de la trama en el momento de su recepción, por lo tanto, se incrementa justo antes de enviarse. Este campo permite controlar la dispersión de los mensajes.

Tabla 5.3: Campos de la trama RREQ del protocolo AODV-LAB

Para una mejor comprensión de la trama y sus campos, a continuación se muestra un ejemplo comentado.

```
[NNID|frameID|SNID|    seqN    |DNID|PNID|hops]
[ 0   |    Q   | S   |1304427351| D   | S   | 1   ]
```

El campo NNID a cero indica que esta trama se ha enviado en modo broadcast. Por ello, ningún nodo que la reciba la descartará a priori. Únicamente lo harán si debe ser así tras su procesado. El campo `frameID` con el carácter `Q` nos informa que se trata de un mensaje RREQ. Esta solicitud de ruta la ha iniciado el SNID, es decir, el nodo `S`. Su número de secuencia, `seqN`, es 1304427351. El campo DNID se corresponde con el nodo con el que se quiere comunicar SNID, en este caso el `D`. PNID es el identificador del nodo inmediato del que se recibe esta trama y su valor aquí coincide con SNID. Esto quiere decir que este mensaje se recibe directamente desde el nodo que lo genera, sin pasar por ningún nodo repetidor, aunque podría perfectamente ser así. Los nodos contenidos en el campo PNID de los mensajes recibidos son vecinos. Por último, `hops` contiene el número de saltos que ha dado el mensaje previamente a su recepción. En el ejemplo, su valor es el mínimo posible, es decir, uno. Esto indica que el mensaje se ha recibido directamente del nodo que lo generó.

### 5.3.3.2. Trama RREP

Se describe en la Tabla 5.4.

Campos de la trama RREP	Identificador	Definición	Descripción
Next Node ID	NNID	<code>char nnid[NID_BYTES];</code>	Indica el destinatario inmediato de la trama, es decir, el próximo nodo en la ruta hacia el destinatario final.
Frame ID	frameID	<code>char frameID[FID_BYTES];</code>	Indica el tipo de trama.
Previous Node ID	PNID	<code>char pnid[NID_BYTES];</code>	Indica el predecesor inmediato en la ruta, es decir, al nodo del que hemos recibido la trama. El nodo fuente se pone a sí mismo como predecesor.
Source Node ID	SNID	<code>char snid[NID_BYTES];</code>	Indica el nodo que inició la respuesta de solicitud de ruta. Se corresponde con el campo DNID del mensaje RREQ al que responde. En esta aplicación coincidirá con el nodo colector.
Sequence Number	seqN	<code>unsigned long int seqN;</code>	Número de secuencia. Se utiliza para evitar bucles e identificar la información más reciente.
Hop Counter	HOPS	<code>int hops;</code>	Indica el número de saltos en la ruta de la trama en el momento de su recepción, por lo tanto, se incrementa justo antes de enviarse. Este campo permite controlar la dispersión de los mensajes.
Destination Node ID	DNID	<code>char dnid[NID_BYTES];</code>	Indica el nodo destino de la respuesta de solicitud de ruta. Se corresponde con el campo SNID del mensaje RREQ al que responde.

Tabla 5.4: Campos de la trama RREP del protocolo AODV-LAB

Para una mejor comprensión de la trama y sus campos, a continuación se muestra un ejemplo comentado.

```
[NNID|frameID|PNID|SNID|    seqN    |hops|DNID]
[ S   |     P   |   A   |   D   |1304433774| 2   | S   ]
```

Al haberse formado el camino de vuelta con un mensaje RREQ, su respuesta se realiza de modo unicast, por ello, el campo NNID en los mensajes RREP lleva el identificador de un nodo concreto, en este caso el S. El frameID es la letra P, correspondiente a este tipo de mensajes. El campo PNID señala que el nodo A es el nodo anterior inmediato en la ruta, es decir, del que se recibe la trama. El nodo D, indicado en SNID, se corresponde con el nodo que generó el RREP, por consiguiente, coincide con el campo DNID del mensaje RREQ al que responde. En este caso, el número de secuencia, seqN, es 1304433774. El número de saltos, hops, es dos. Concretamente del nodo D al A, y éste que se lo envía al S. Por último, el campo DNID, que indica el destinatario del RREP y se corresponde con el SNID del mensaje RREQ, es el nodo S. Finalmente se deduce que cuando los campos NNID y DNID coinciden el mensaje ha llegado a su destinatario final, de manera que, se habrá creado la correspondiente ruta bidireccional entre dos nodos.

### 5.3.3.3. Trama DATA

Se describe en la Tabla 5.5.

Campos de la trama DATA	Identificador	Definición	Descripción
Next Node ID	NNID	char nnid[NID_BYTES];	Indica el destinatario inmediato de la trama, es decir, el próximo nodo en la ruta hacia el destinatario final.
Frame ID	frameID	char frameID[FID_BYTES];	Indica el tipo de trama.
Destination Node ID	DNID	char dnid[NID_BYTES];	Indica el nodo destino de la trama DATA. En esta aplicación coincidirá siempre con el nodo colector.
Source Node ID	SNID	char snid[NID_BYTES];	Indica el nodo que manda los datos. En esta aplicación serán siempre alarmas.
Alarm ID	alarmID	char alarmID[11];	Junto con el campo SNID identifica inequívocamente cada alarma. Cada nodo va incrementando este valor según genera nuevas alarmas.
Timestamp	timeStamp	char timeStamp[11];	Timestamp (fecha y hora) en que se generó la alarma.
Alarm type	alarmType	char alarmType[ATYPE_BYTES];	Indica el tipo de alarma generada en función del evento acústico registrado.
Latitude	latitude	char latitude[MAXALEN];	Latitud geográfica del nodo en el momento de generar la alarma. Se obtiene de un GPS.
Longitude	longitude	char longitude[MAXALEN];	Longitud geográfica del nodo en el momento de generar la alarma. Se obtiene de un GPS.
Percentage of confidence	perConf	char perConf[MAXALEN];	Porcentaje de confianza asociado a la alarma generada.

Tabla 5.5: Campos de la trama DATA del protocolo AODV-LAB

Para una mejor comprensión de la trama y sus campos, a continuación se muestra un ejemplo comentado.

```
[NNID|frameID|DNID|SNID|alarmID| timeStamp |alarmType|latitude|longitude|perConf]
[ A | D | D | S | 1 | 1304433727 | W | 41.2061 | 1.7300 | 87 ]
```

La trama se envía al nodo A. Es de tipo DATA ya que su **frameID** es la letra D. El mensaje va del nodo S al D como indican respectivamente los campos **SNID** y **DNID**. El identificador de alarma, **alarmID** es uno, lo que señala que es la primera alarma que genera el nodo S. El campo **timeStamp**, con valor 1304433727, nos permite conocer la fecha y hora en que se generó la alarma. Concretamente, en el sistema operativo GNU/Linux se utiliza el timestamp UNIX, que es el número de segundos desde 00:00:00 UTC del 1 de enero de 1970 hasta el momento en que se crea el timestamp. De esta manera, utilizando un conversor, se puede obtener que esta alarma se generó el 3 de mayo de 2011 a las 15:42:07 (GMT+1). Como se indica en la Tabla 5.5, el campo **alarmType** determina el tipo de alarma en función del evento acústico registrado. En este proyecto no se ha definido un diccionario de alarmas, así que, se ha utilizado la letra W únicamente como ejemplo. En el momento de generar la alarma el nodo se encontraba en las coordenadas geográficas 41.2061 N - 1.73 E. Los valores negativos corresponderán a la latitud sur (S) y longitud oeste (W). Por último, el porcentaje de confianza de la alarma generada es del 87 %, como indica el campo **perConf**.

#### 5.3.3.4. Trama ACK

Se describe en la Tabla 5.6.

Campos de la trama ACK	Identificador	Definición	Descripción
Next Node ID	NNID	<code>char nnid[NID_BYTES];</code>	Indica el destinatario inmediato de la trama, es decir, el próximo nodo en la ruta hacia el destinatario final.
Frame ID	frameID	<code>char frameID[FID_BYTES];</code>	Indica el tipo de trama.
Destination Node ID	DNID	<code>char dnid[NID_BYTES];</code>	Indica el nodo destino de la trama ACK. Se corresponderá con el campo SNID de la alarma a la que responde.
Alarm ID	alarmID	<code>char alarmID[11];</code>	Identificador de alarma. Cuando un nodo recibe un ACK y él es el DNID de la trama, con el campo alarmID puede saber que alarma de las que envió se entregó con éxito.

Tabla 5.6: Campos de la trama ACK del protocolo AODV-LAB

Para una mejor comprensión de la trama y sus campos, a continuación se muestra un ejemplo comentado.

```
[NNID|frameID|DNID|alarmID]
[ S |   A   | S   |   1   ]
```

La trama se envía al nodo S, que además es el destinatario final de la misma, como indican los campos NNID y DNID. Es una trama ACK ya que su frameID es la letra A. Concretamente este acuse de recibo le indica al nodo S que su primera alarma generada y enviada (alarmID = 1) se ha recibido con éxito.

### 5.3.3.5. Trama PREQ

Se describe en la Tabla 5.7.

Campos de la trama PREQ	Identificador	Definición	Descripción
Next Node ID	NNID	<code>char nnid[NID_BYTES];</code>	Indica el destinatario inmediato de la trama, es decir, el próximo nodo en la ruta hacia el destinatario final. El valor 0 se corresponde con el modo Broadcast.
Frame ID	frameID	<code>char frameID[FID_BYTES];</code>	Indica el tipo de trama.
Source Node ID	SNID	<code>char snid[NID_BYTES];</code>	Indica el nodo que inició la solicitud de posición. En esta aplicación coincidirá con el nodo colector.
Sequence Number	seqN	<code>unsigned long int seqN;</code>	Número de secuencia. Se utiliza para evitar bucles e identificar la información más reciente. La trama PREQ descubre rutas de vuelta hacia el nodo que la envió, por eso incluye este campo.
Previous Node ID	PNID	<code>char pnid[NID_BYTES];</code>	Indica el predecesor inmediato en la ruta, es decir, el nodo del que hemos recibido la trama. El nodo fuente se pone a sí mismo como predecesor.
Hop Counter	HOPS	<code>int hops;</code>	Indica el número de saltos en la ruta de la trama en el momento de su recepción, por lo tanto, se incrementa justo antes de enviarse. En los mensajes PREQ no es un factor para controlar la dispersión ya que una solicitud de posición debe llegar a todos los nodos de la red.

Tabla 5.7: Campos de la trama PREQ del protocolo AODV-LAB

Para una mejor comprensión de la trama y sus campos, a continuación se muestra un ejemplo comentado.

```
[NNID|frameID|SNID| seqN |PNID|hops]
[ 0 |   W | D |1304516600| D | 1 ]
```

El nodo D envía un mensaje PREQ en modo broadcast. El número de secuencia en el mensaje es 1304516600. En este caso, el nodo predecesor, PNID, se corresponde con el SNID, por lo tanto, la trama capturada únicamente ha dado un salto, como se puede observar en el campo hops del ejemplo.

### 5.3.3.6. Trama PREP

Se describe en la Tabla 5.8.

Campos de la trama PREP	Identificador	Definición	Descripción
Next Node ID	NNID	char nnid[NID_BYTES];	Indica el destinatario inmediato de la trama, es decir, el próximo nodo en la ruta hacia el destinatario final.
Frame ID	frameID	char frameID[FID_BYTES];	Indica el tipo de trama.
Destination Node ID	DNID	char dnid[NID_BYTES];	Indica el nodo destino de la respuesta de solicitud de posición. Se corresponde con el campo SNID del mensaje PREQ al que responde, que en esta aplicación siempre será el nodo colector.
Timestamp	timeStamp	char timeStamp[11];	Timestamp (fecha y hora) en que se genera el PREP.
Source Node ID	SNID	char snid[NID_BYTES];	Indica el nodo que inició la respuesta de solicitud de posición, es decir el que está indicando sus coordenadas geográficas en esta misma trama.
Latitude	latitude	char latitude[MAXALEN];	Latitud geográfica del nodo en el momento de generar el PREP. Se obtiene de un GPS.
Longitude	longitude	char longitude[MAXALEN];	Longitud geográfica del nodo en el momento de generar el PREP. Se obtiene de un GPS.

Tabla 5.8: Campos de la trama PREP del protocolo AODV-LAB

Para una mejor comprensión de la trama y sus campos, a continuación se muestra un ejemplo comentado.

```
[NNID|frameID|DNID| timeStamp|SNID|latitude|longitude]
[ D |   C | D |1304516621| A | 41.2061 | 1.7300 ]
```

Esta trama PREP, con frameID C, va dirigida al nodo D, que además fué el que inició la solicitud de posiciones con un mensaje PREQ. El timestamp que contiene la fecha y hora en la que el nodo A consulta su posición al GPS es 1304516621. Por último, los campos SNID, latitude y longitude indican que en el momento en que se generó el PREP el nodo A se encontraba en las coordenadas geográficas 41.2061 N - 1.73 E.

### 5.3.4. Estructura y flujo de ejecución del programa

En la Figura 5.7 se muestra la estructura del programa principal, donde se pueden diferenciar claramente dos etapas, una primera que realiza una serie de operaciones de manera secuencial y, la segunda que lo hace en bucle indefinidamente hasta que se detiene la ejecución del programa.

En la primera parte se realizan las siguientes operaciones:

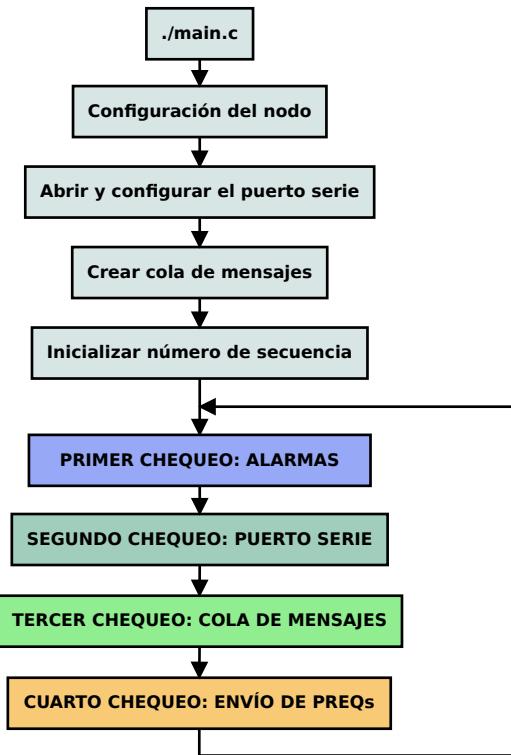


Figura 5.7: Estructura y flujo de ejecución del programa.

■ **Configuración del nodo.** Lectura del fichero de configuración `node.conf` donde se definen los siguientes parámetros:

- **SERIAL\_NAME.** Nombre del puerto serie en el sistema. En GNU/Linux serán de la siguiente manera: `/dev/ttyUSB0`.
- **SERIAL\_BAUD\_RATE.** Velocidad del puerto serie en bits por segundo (bps). En el fichero `node.conf` se indican los valores posibles.
- **NODE\_ID.** Identificador del propio nodo.
- **SINK\_NODE\_ID.** Identificador del nodo colector.
- **MAX\_NUM\_HOPS.** Número máximo de saltos en una ruta.
- **LIFETIME\_RTENTRY.** Tiempo de vida de las entradas de la tabla de rutas. Tras este periodo las rutas dejan de ser válidas.
- **ALARM\_RETRIES.** Número de reintentos de envío de una alarma en caso de que no se entregue con éxito.
- **ALARM\_TIMEOUT.** Tiempo de espera tras el envío de una alarma para recibir su correspondiente acuse de recibo (ACK). Si en ese periodo no llega, se procederá al reenvío de la alarma.
- **RREQ\_TIMEOUT.** Tiempo de espera tras el envío de una solicitud de ruta (RREQ) para recibir su correspondiente respuesta (RREP). Si en ese periodo no llega, se procederá al reenvío del RREQ.

- **Abrir y configurar el puerto serie**, utilizando los valores leídos anteriormente en el fichero `node.conf`.
- **Crear cola de mensajes**. Ésta seguirá la disciplina FIFO. En ella se guardarán los mensajes que se reciban por el puerto serie, y se tomarán posteriormente para ser procesados.
- **Inicializar número de secuencia**. El número de secuencia se inicializa siempre que se ejecuta el programa con la función `time()`. De esta manera, aunque un nodo se resetee, al reiniciarse no causará ningún problema en la red, ya que los números de secuencia que utilizará serán mayores que los empleados antes del reseteo.

Estas operaciones son imprescindibles para que el nodo pueda operar en la red correctamente. En caso de que alguna de ellas falle, se detendrá la ejecución del programa automáticamente, indicando por pantalla el motivo de error para facilitar su corrección.

En la segunda parte del programa se realizan en bucle cuatro procesos distintos. A continuación, se detalla cada uno de ellos.

#### 5.3.4.1. Primer chequeo: Alarmas

El proceso de chequeo de alarmas se encarga de gestionar: la recepción de acuses de recibo, el envío y reenvío alarmas y, en caso de ser necesario, el inicio del descubrimiento de ruta hacia el nodo colector. En la Figura 5.8 se muestra el diagrama de flujo de este proceso. En él se puede observar que operaciones se van realizando según las condiciones actuales del sistema como: estado de espera de acuse de recibo, contador de reintentos de envío de alarmas, caminos válidos disponibles en la tabla de rutas, etc.

#### 5.3.4.2. Segundo chequeo: Puerto serie

En segundo lugar, se realiza el chequeo del puerto serie. Como se observa en la Figura 5.9, inicialmente se lee el puerto durante `SERIAL_TIMEOUT`, definido en el programa. Si no hay nada de información se finaliza el proceso. En caso contrario, se buscan tramas completas entre los datos recibidos y, si se encuentra alguna, se coloca en la cola de mensajes. Si, a pesar de haber recibido información por el puerto serie, no se detecta ninguna trama completa esta se descarta y se finaliza el chequeo.

#### 5.3.4.3. Tercer chequeo: Cola de mensajes

En esta parte del programa principal se leen todos los mensajes recibidos y almacenados en la cola de mensajes hasta ese momento. Esto es así por dos razones:

- Cuando se procesa un mensaje se actualiza la información del propio sistema, del estado de la red, etc. Por ello, cada vez que se llega a esta parte del programa principal, interesa procesar todos los mensajes actualizando así al máximo dicha información, que será utilizada en otros chequeos, como el de alarmas. Esto evita el envío de mensajes innecesarios, cosa que podría ocurrir si al pasar por el chequeo de la cola de mensajes solo se procesase un mensaje por pasada.

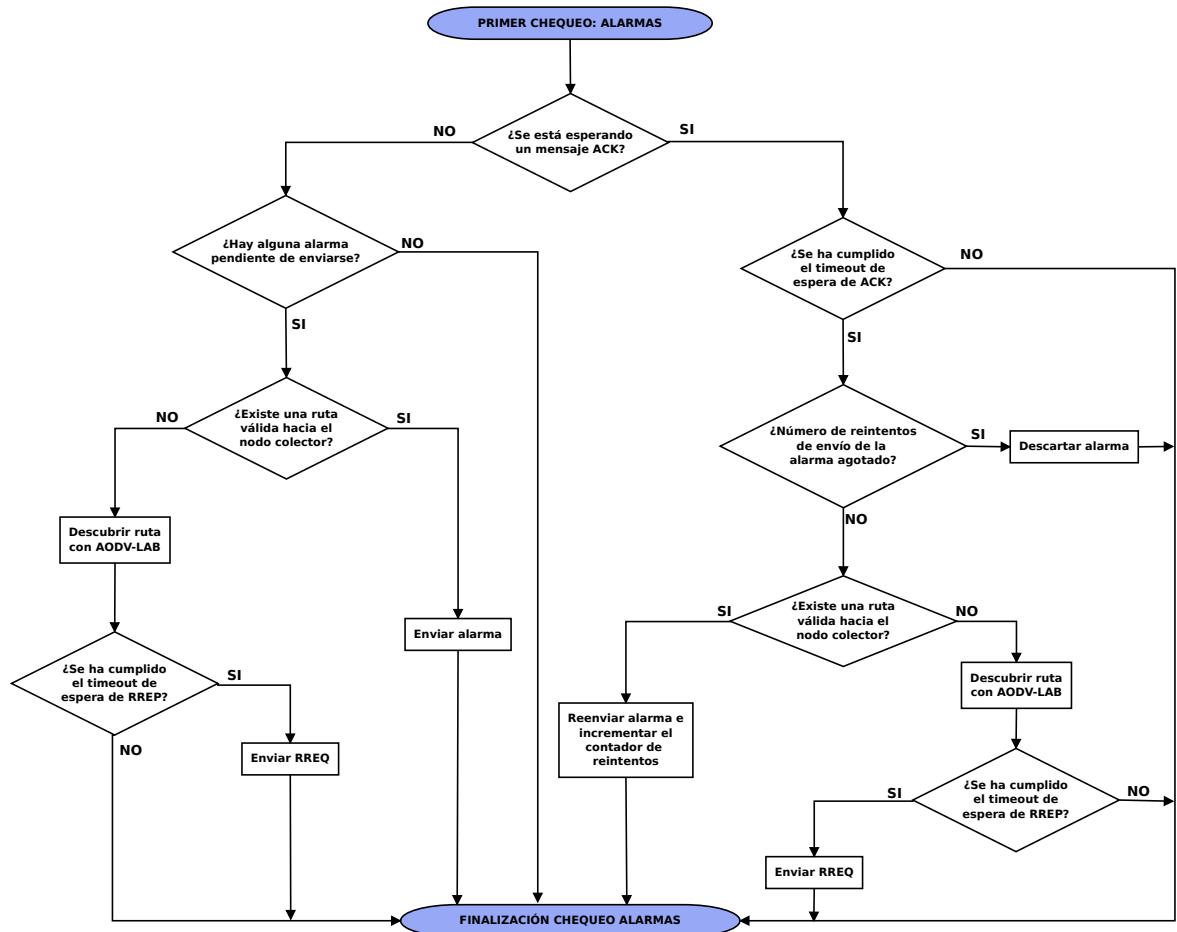


Figura 5.8: Diagrama de flujo de la parte de chequeo de alarmas.

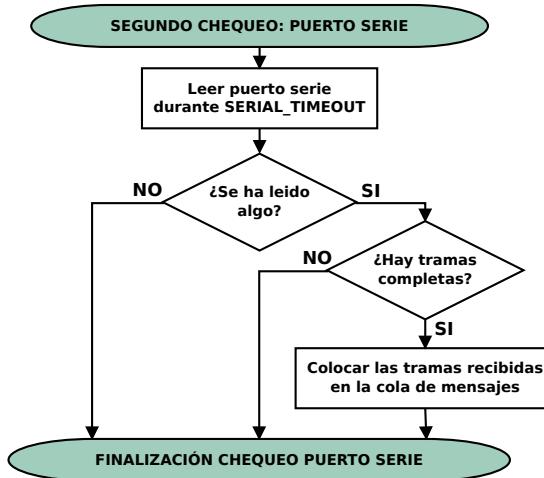


Figura 5.9: Diagrama de flujo de la parte de chequeo del puerto serie.

- Cada vez que se finaliza esta acción la cola de mensajes queda vacía, evitando problemas de almacenamiento de datos.

Como se observa en la Figura 5.10, cuando se toma un mensaje de la cola, el nodo se encarga de comprobar si es el destinatario del mismo, o si éste fue enviado en modo broadcast. Si se da uno de estos dos casos, existe una función encargada del procesamiento de cada cada tipo de mensaje. Así, dependiendo de su campo `frameID`, el mensaje es enviado a la función correspondiente para su procesado. Al terminar esta operación, o tras haber descartado un mensaje, se vuelve a comprobar si hay más en la cola y, únicamente cuando esta está vacía, se termina el proceso.

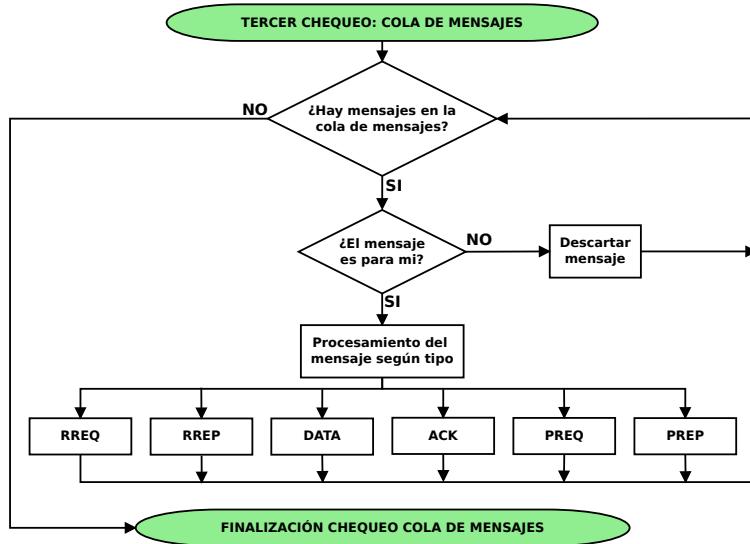


Figura 5.10: Diagrama de flujo de la parte de chequeo de la cola de mensajes.

#### 5.3.4.4. Cuarto chequeo: Envío de PREQ

Como se comentó en la Subsección 5.1.2, el nodo colector tiene la posibilidad de enviar mensajes de solicitud de posición en modo broadcast para conocer la localización exacta del resto de nodos de la red. De la gestión de esta operación se encarga el proceso de chequeo de envío de mensajes PREQ. Como se muestra en la Figura 5.11, en esta parte del programa el nodo comprueba si el mismo es el colector o no. En caso de serlo, la señal para enviar un mensaje PREQ (*Position Request*) en modo broadcast es la pulsación de la letra w o W, que proviene de la sentencia *Where are you?*. Si esta tecla se ha pulsado, al llegar a este punto el nodo enviará un PREQ y finalizará este cuarto chequeo, volviendo nuevamente al primero, como se mostraba en la Figura 5.7. Incluir esta parte en el programa permite emplear un código genérico para todos los nodos, independientemente de ser o no colector, utilizando el archivo de configuración para definir estas características.

## 5.4. Validación y resultados

Para verificar el correcto funcionamiento del software implementado se ha realizado una simulación del sistema completo. Es importante destacar que un sistema de estas características en funcionamiento puede dar lugar a un gran número de situaciones distintas, que vendrán determinadas por: el flujo de alarmas, el número de nodos de la red,

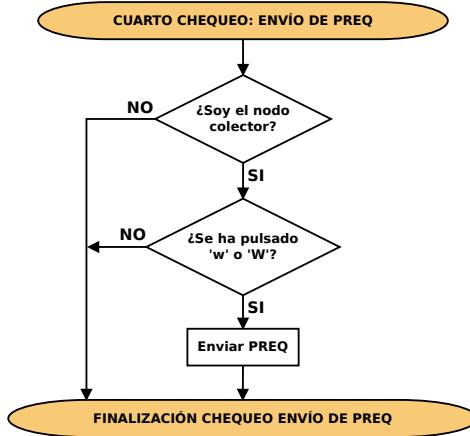
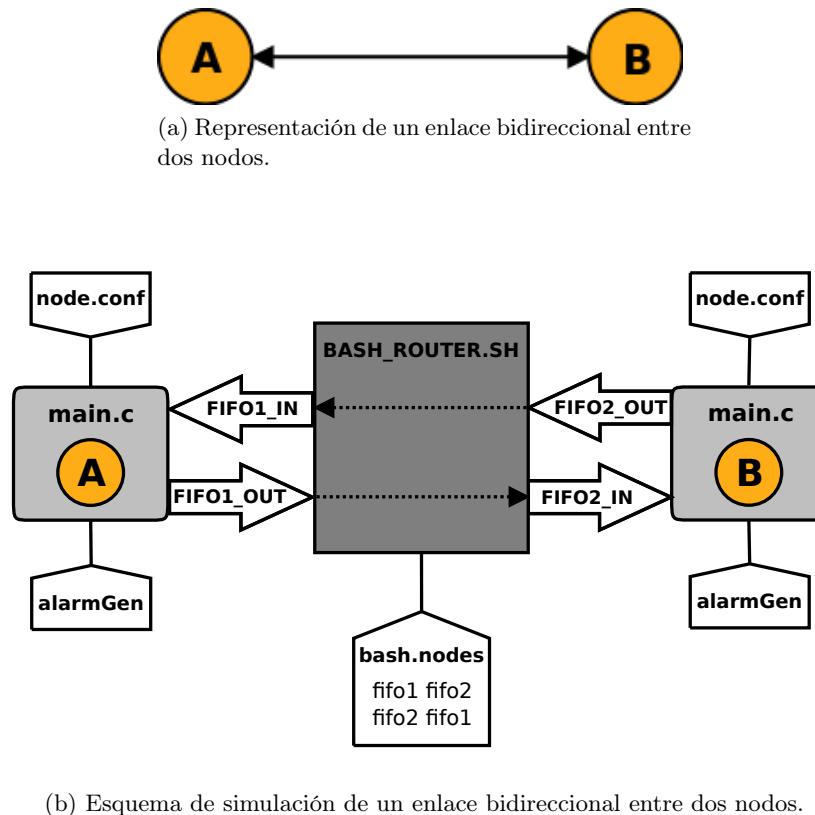


Figura 5.11: Diagrama de flujo de la parte de envío de mensajes PREQ.

los parámetros de configuración u otras circunstancias. Contemplar todos estos casos en una simulación es una tarea larga y compleja que, para la implementación de una primera versión o prototipo, no tiene lugar. Por ello, esta sección únicamente pretende constatar las principales funcionalidades deseadas, tanto del protocolo de encaminamiento AODV-LAB, como del módulo de entrega remota de datos. Concretamente se han simulado cuatro escenarios, cada uno de ellos con una distribución distinta, en número y posición, de nodos. A continuación, se detalla el procedimiento seguido para la simulación, así como las características de cada escenario y los resultados obtenidos.

#### 5.4.1. Método de simulación

Emular el funcionamiento real del sistema completo desarrollado implica, por un lado, simular los nodos de la red operando independientemente y, por otro, los enlaces entre ellos. Las Figuras 5.12 y 5.13 representan un esquema global del método utilizado.



(b) Esquema de simulación de un enlace bidireccional entre dos nodos.

Figura 5.12: Simulación de un enlace bidireccional entre dos nodos.

Cada nodo del sistema en la simulación consta principalmente de tres partes:

- Fichero de configuración (`node.config`). Contiene los parámetros de configuración del nodo, por ejemplo: identificador del propio nodo, identificador del nodo colector, número máximo posible de saltos en una ruta, etc. (Véase la Subsección 5.3.4).
- Generador aleatorio de alarmas (`alarmGen`). Cada cierto intervalo de tiempo, de duración variable entre dos márgenes definidos, se forma una alarma y se incluye en el fichero `alarm.dat`, que si no existe previamente se crea. Cada alarma lleva su correspondiente identificador que la diferencia inequívocamente de las demás.
- Programa principal (`main`). Consiste en el módulo de entrega remota de datos descrito en la Sección 5.1, que a su vez utiliza el protocolo de encaminamiento AODV-LAB, descrito en la Subsección 5.2.4.

Durante la simulación, los distintos nodos existentes se ejecutan en paralelo en la misma máquina, en este caso bajo el sistema operativo GNU/Linux Ubuntu 10.10 64 bits.

Por otro lado, se debe definir el modo en que los nodos se comunicarán entre sí. En la realidad, dos nodos podrán comunicarse cuando se encuentren respectivamente uno dentro del área de cobertura del otro. En la simulación, esto se simplifica y se definen directamente los enlaces entre los nodos. Esto tiene lugar en el fichero `bash.nodes`, donde en la primera columna aparecen listados todos los nodos de la simulación y a la derecha de cada uno la lista de nodos con los que puede comunicarse. Por ejemplo,

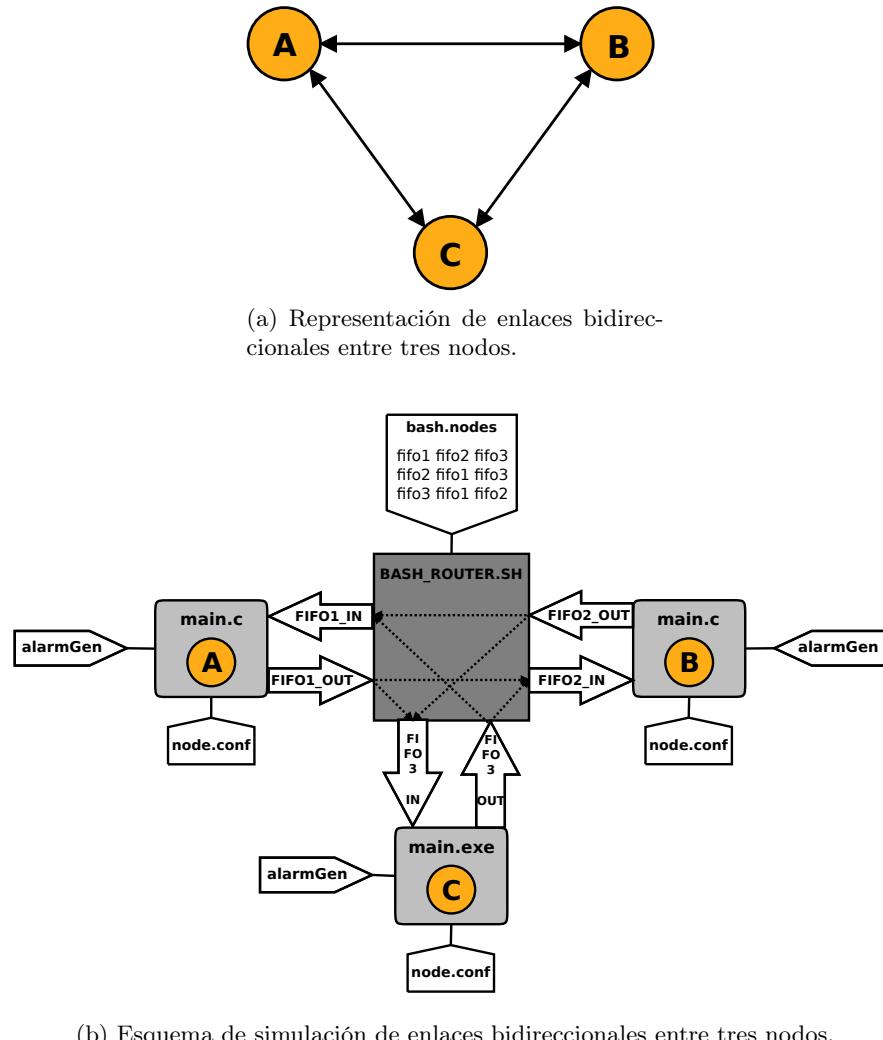


Figura 5.13: Simulación de enlaces bidireccionales entre tres nodos.

el siguiente fragmento nos indican que el nodo 1 puede comunicarse con el 2, éste a su vez con el nodo 1 y 3 y, por último, el nodo 3 únicamente con el nodo 2.

Ejemplo fichero `bash.nodes`:

```
fifo1 fifo2
fifo2 fifo1 fifo3
fifo3 fifo2
```

Se emplea la notación `fifoX`, donde X es el número correspondiente al nodo, ya que la información se escribe y lee en ficheros tipo FIFO. Concretamente se utilizan ficheros `fifoX_out` donde los nodos escriben la información que transmiten y `fifoX_in` donde leen la que reciben. Para crear estos ficheros desde consola basta con introducir el comando: `$ mkfifo fifoX_out`. Para que los nodos trabajen en modo simulación, es decir utilizando los ficheros FIFO, en vez de en modo de operación, que emplea el puerto serie, debe ponerse la variable global `SIMULATION` en `serial.c` a uno.

Por último, se ha implementado un script en bash, `bash_router.sh`, que se encarga de leer periódica y secuencialmente los ficheros `fifoX_out` y en caso de encontrar algún mensaje escrito en alguno de ellos lo copia en los ficheros `fifoX_in` correspondientes

según esté indicado en `bash.nodes`.

Este método de simulación se comporta como si se trabajase en un medio ideal, sin interferencias ni ruido. Para emular una rotura de enlace se procederá deteniendo la ejecución del `main` de algún nodo. Esto se tratará con mayor detalle en las próximas subsecciones.

Además, se ha programado el script `bash_router.sh` para que imprima por pantalla los mensajes que se escriben en cada fichero `fifoX_out`. De esta manera, y apollándose en el formato de las tramas descrito en la Subsección 5.3.3, se comprenderá mejor el modo de operación de la red. También se utilizarán los ficheros `alarmReg.dat` que genera el nodo colector, `nodo D` en todos los ejemplos, para registrar las alarmas que recibe. Así, se tendrá constancia del orden de llegada de estas, o de si llegan repetidas. Se procederá de igual manera con el fichero `positionReg.dat` cuando sea necesario. Por último, en algunos casos, cuando esto clarifique la explicación de lo ocurrido durante la simulación, se utilizarán fragmentos de la información que van reportando los nodos durante su operación en la red.

#### 5.4.2. Escenario 1

El primer escenario a simular es el caso más simple posible. Como se observa en la Figura 5.14, se trata de un enlace entre dos nodos, donde el `nodo S` sería un nodo generador de alarmas y el `nodo D` un nodo colector.

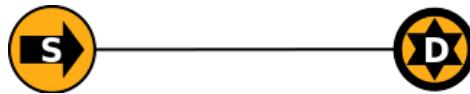


Figura 5.14: Escenario 1 de simulación:  
Enlace bidireccional entre dos nodos.

A continuación, se detallan las diferentes situaciones simuladas sobre este escenario y se comentan los resultados obtenidos. Destacar que `fifo2_out` es lo que transmite el `nodo S` y `fifo1_out` lo del `nodo D`.

##### 1. Descubrimiento de nodos vecinos, envío de alarmas y acuses de recibo.

El `nodo S` tiene dos alarmas pendientes de enviar.

```

1 // El nodo S envía en modo broadcast un RREQ buscando una
2 // ruta hacia el nodo D
3 fifo2_out: [0|Q|S|1304421715|D|S|1]                                // RREQ
4 // Cuando el nodo D recibe el RREQ, guarda la ruta hacia
5 // el nodo S y responde en modo unicast con un RREP
6 fifo1_out: [S|P|D|D|1304421716|1|S]                                // RREP
7 // Tras recibir el nodo S el RREP, conoce una ruta hacia
8 // el nodo D y envía la primera alarma
9 fifo2_out: [D|D|D|S|1|1304421690|W|41.2061|1.7300|87]    // ALARM1
10 // Cuando el nodo D recibe la alarma envía un acuse de
11 // recibo hacia el nodo S
12 fifo1_out: [S|A|S|1]                                              // ACK1
13 // Sólo cuando el nodo S recibe el acuse de recibo de la
14 // alarma anterior, procede a enviar la siguiente
15 fifo2_out: [D|D|D|S|2|1304421694|W|41.2061|1.7300|87]    // ALARM2
16 // Al recibir el nodo D la alarma, envía un acuse de

```

```

17 // recibo hacia el nodo S
18 fifo1_out: [S|A|S|2] // ACK2
19 // El nodo S recibe el acuse de recibo y como no tiene más
20 // alarmas pendientes, por el momento no envía nada más.

```

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304421690	W	41.2061	1.7300	87
2	1304421694	W	41.2061	1.7300	87

Destacar que los campos `alarmType`, `latitude`, `longitude` y `perConf` de las alarmas que genera el `alarmGen` no varían, únicamente varían el `alarmID` y el `timeStamp`.

2. **Falla la comunicación temporalmente entre los dos nodos, pero ninguno se resetea.** El nodo S tiene dos alarmas pendientes de enviar.

```

1 // El nodo S envía en modo broadcast un RREQ buscando una
2 fifo2_out: [0|Q|S|1304423364|D|S|1] // RREQ
3 fifo1_out: [S|P|D|D|1304423365|1|S] // RREP
4 fifo2_out: [D|D|D|S|1|1304423354|W|41.2061|1.7300|87] // ALARM1
5 // Tras ALARM_TIMEOUT sin recibir el correspondiente ACK y,
6 // como ALARM_RETRIES es 2, el nodo S vuelve a intentar
7 // enviar su alarma
8 fifo2_out: [D|D|D|S|1|1304423354|W|41.2061|1.7300|87] // ALARM1
9 // Esta vez llega correctamente al nodo D y éste responde
10 // con el correspondiente acuse de recibo
11 fifo1_out: [S|A|S|1] // ACK1
12 fifo2_out: [D|D|D|S|2|1304423359|W|41.2061|1.7300|87] // ALARM2
13 fifo1_out: [S|A|S|2] // ACK2

```

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304423354	W	41.2061	1.7300	87
1	1304423354	W	41.2061	1.7300	87
2	1304423359	W	41.2061	1.7300	87

Al contemplar el fichero de registro de alarmas se ve que la primera está repetida. Esto indica que la alarma llegó perfectamente al nodo D y que la comunicación falló exáctamente en el envío del acuse de recibo. Por ello, el nodo S al no recibirla desconoce que su alarma ha llegado correctamente y, tras comprobar que en su configuración se ha fijado el número de reintentos a dos, la vuelve a enviar.

3. Agotar los reintentos con la última alarma del `alarm.dat`. El nodo S tiene dos alarmas pendientes de enviar.

```

1 fifo2_out: [0|Q|S|1304424109|D|S|1]                                // RREQ
2 fifo1_out: [S|P|D|D|1304424110|1|S]                                // RREP
3 fifo2_out: [D|D|D|S|1|1304424100|W|41.2061|1.7300|87]      // ALARM1
4 fifo1_out: [S|A|S|1]                                              // ACK1
5 fifo2_out: [D|D|D|S|2|1304424104|W|41.2061|1.7300|87]      // ALARM2
6 // Tras ALARM_TIMEOUT, primer reinicio
7 fifo2_out: [D|D|D|S|2|1304424104|W|41.2061|1.7300|87]      // ALARM2
8 // Tras ALARM_TIMEOUT segundo y último reinicio
9 fifo2_out: [D|D|D|S|2|1304424104|W|41.2061|1.7300|87]      // ALARM2
10 // Descarta la alarma y, como no hay nuevas alarmas pendientes,
11 // no realiza ninguna operación más

```

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304424100	W	41.2061	1.7300	87

Tras enviar una alarma correctamente, al tratar de enviar la segunda y última pendiente en el fichero `alarm.dat`, no recibe respuesta del nodo D. Efectúa el número de reintentos definido por `ALARM_RETRIES` y una vez se agotan, al ser la última pendiente, la descarta y no realiza más operaciones. En el fichero `alarmReg.dat` generado por el nodo D únicamente aparece la primera alarma. Esto indica que la comunicación falló antes de que este nodo recibiese la segunda y probablemente no se volvió a recuperar. Esto podría corresponderse con una situación en la realidad en la que el nodo colector sale del área de cobertura de un nodo que le estaba enviando alarmas. Al proceder como se ha descrito se evita que el nodo continúe enviando mensajes sin sentido y por lo tanto malgastnado recursos. La alarma se descarta porque tras dos reintentos habrá transcurrido tres veces el tiempo definido en `ALARM_TIMEOUT`, por lo cual, la información de dicha alarma será anticuada.

4. Agotar los reintentos con una alarma y tener otras posteriores pendientes en el `alarm.dat`. El nodo S tiene tres alarmas pendientes de enviar.

```

1   fifo2_out: [0|Q|S|1304424294|D|S|1]                                // RREQ
2   fifo1_out: [S|P|D|D|1304424295|1|S]                                // RREP
3   fifo2_out: [D|D|D|S|1|1304424257|W|41.2061|1.7300|87]           // ALARM1
4   fifo1_out: [S|A|S|1]                                              // ACK1
5   fifo2_out: [D|D|D|S|2|1304424260|W|41.2061|1.7300|87]           // ALARM2
6   // Tras ALARM_TIMEOUT, primer reintento
7   fifo2_out: [D|D|D|S|2|1304424260|W|41.2061|1.7300|87]           // ALARM2
8   // Tras ALARM_TIMEOUT segundo y último reintento
9   fifo2_out: [D|D|D|S|2|1304424260|W|41.2061|1.7300|87]           // ALARM2
10  // Descarta la alarma 2 y como todavía tiene una tercera
11  // pendiente inicia un Descubrimiento de Ruta para tratar
12  // de encontrar un camino válido al nodo D, ya que el que
13  // tenía no le ha permitido entregar la alarma anterior.
14  fifo2_out: [0|Q|S|1304424297|D|S|1]                                // RREQ
15  fifo1_out: [S|P|D|D|1304424347|1|S]                                // RREP
16  // Descubre una ruta, que puede ser la misma de antes
17  // o no, y envía la siguiente alarma pendiente
18  fifo2_out: [D|D|D|S|3|1304424267|W|41.2061|1.7300|87]           // ALARM3
19  fifo1_out: [S|A|S|3]                                              // ACK3
20  // El nodo S no tiene más alarmas que entregar

```

Fichero alarmReg.dat generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304424257	W	41.2061	1.7300	87
3	1304424267	W	41.2061	1.7300	87

Esta situación es similar a la anterior, pero esta vez, después de la alarma descartada, hay otra pendiente de ser enviada. Por ello, el nodo S reinicia el proceso de Descubrimiento de Ruta. Si éste concluye con éxito, como es el caso, continúa mandando las alarmas por la nueva ruta, que podría ser la misma que la que falló. En el fichero `alarmReg.dat` se puede ver que únicamente han quedado registradas la primera y última alarma.

5. Tratar periódicamente de descubrir una ruta hacia el nodo colector cuando hay alarmas pendientes de ser enviadas. El nodo S tiene dos alarmas pendientes de enviar.

```

1 fifo2_out: [0|Q|S|13044424779|D|S|1] // RREQ1
2 // Tras RREQ_TIMEOUT se vuelve a enviar un RREQ
3 fifo2_out: [0|Q|S|13044424780|D|S|1] // RREQ2
4 // Tras RREQ_TIMEOUT se vuelve a enviar un RREQ
5 fifo2_out: [0|Q|S|13044424781|D|S|1] // RREQ3
6 // Tras RREQ_TIMEOUT se vuelve a enviar un RREQ
7 fifo2_out: [0|Q|S|13044424782|D|S|1] // RREQ4
8 // Tras RREQ_TIMEOUT se vuelve a enviar un RREQ
9 fifo2_out: [0|Q|S|13044424783|D|S|1] // RREQ5
10 fifo1_out: [S|P|D|D|13044424811|1|S] // RREP
11 fifo2_out: [D|D|D|S|1|13044424753|W|41.2061|1.7300|87] // ALARM1
12 fifo1_out: [S|A|S|1] // ACK1
13 fifo2_out: [D|D|D|S|2|13044424756|W|41.2061|1.7300|87] // ALARM2
14 fifo1_out: [S|A|S|2] // ACK2

```

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304424753	W	41.2061	1.7300	87
2	1304424756	W	41.2061	1.7300	87

El Descubrimiento de Ruta, al contrario que el envío de una alarma, no tiene un número de reintentos definido. Mientras haya alarmas pendientes de ser enviadas, el sistema tratará de descubrir una ruta hacia el nodo colector periódicamente cada `RREQ_TIMEOUT`. Una vez que encuentre una ruta, enviará las alarmas que tenga acumuladas en su fichero `alarm.dat`.

## 6. Importancia de la inicialización del número de secuencia con la función `time()`. El nodo S tiene dos alarmas pendientes de enviar.

```

1  fifo2_out: [0|Q|S|1304425663|D|S|1]                                // RREQ
2  fifo1_out: [S|P|D|D|1304425664|1|S]                                // RREP
3  fifo2_out: [D|D|D|S|1|1304425637|W|41.2061|1.7300|87]      // ALARM1
4  fifo1_out: [S|A|S|1]                                              // ACK1
5  // El nodo S se resetea y por lo tanto pierde toda
6  // la información que tenía: sus tablas, el identificador
7  // de la última alarma que envió, etc. Tras un periodo
8  // de tiempo se reinicia y como tiene alarmas en el
9  // alarm.dat inicia el Descubrimiento de Ruta.
10 fifo2_out: [0|Q|S|1304425677|D|S|1]                                // RREQ
11 fifo1_out: [S|P|D|D|1304425678|1|S]                                // RREP
12 // Una vez tiene una ruta hacia el nodo D, vuelve a
13 // enviar la primera alarma, ya que en el reseteo perdió
14 // la información de cual fué la última que envió.
15 fifo2_out: [D|D|D|S|1|1304425637|W|41.2061|1.7300|87]      // ALARM1
16 fifo1_out: [S|A|S|1]                                              // ACK1
17 fifo2_out: [D|D|D|S|2|1304425643|W|41.2061|1.7300|87]      // ALARM2
18 fifo1_out: [S|A|S|2]                                              // ACK2

```

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304425637	W	41.2061	1.7300	87
1	1304425637	W	41.2061	1.7300	87
2	1304425643	W	41.2061	1.7300	87

Cuando un nodo se resetea pierde su número de secuencia y la información que guardaba en sus tablas de: RREQ, encaminamiento y PREQ. Por ello, al reiniciarse vuelve a lanzar el proceso de Descubrimiento de Ruta, como se observa en la línea 10. Si sus nodos vecinos no se han reseteado, para que estos al recibir el mensaje RREQ no lo descarten, su número de secuencia debe ser superior al guardado anteriormente en sus tablas de RREQ, que en este caso se corresponde con el del RREQ de la línea 1, es decir, 1304425663. Tal como se veía en la Subsección 5.3.4, al utilizar la función `time()` para inicializar el número de secuencia, en vez de hacerlo con cero, por ejemplo, obtendremos un número mayor que el primero, en este caso 1304425677. La diferencia entre ambos valores, que

es 14, nos indica el número de segundos transcurrido entre las dos inicializaciones. Este número siempre será mayor que el número de mensajes enviados en ese intervalo, por lo tanto, procediendo de esta manera la red soporta sin problemas el reseteo de alguno de sus nodos.

Como vemos en el fichero `alarmReg.dat`, la primera alarma se recibió dos veces, ya que en el reseteo además se perdió el identificador de la última alarma enviada del fichero `alarm.dat`. Este efecto podría ser muy negativo si hubiese muchas alarmas ya enviadas y hubiese que volver a enviarlas pero, por la aplicación que tendrá el servicio de alarmas del sistema LIDO DCL, será muy poco común que se produzca una gran acumulación de estas en un nodo.

7. **Caducidad de las rutas guardadas en la tabla de encaminamiento.** El nodo S tiene inicialmente una alarma pendiente de enviar y, tras un periodo de tiempo superior a `LIFETIME_RTENTRY`, aparecen dos nuevas alarmas que entregar.

```

1 fifo2_out: [0|Q|S|1304427351|D|S|1]                                // RREQ
2 fifo1_out: [S|P|D|D|1304427352|1|S]                                // RREP
3 fifo2_out: [D|D|D|S|1|1304427340|W|41.2061|1.7300|87]      // ALARM1
4 fifo1_out: [S|A|S|1]                                              // ACK1
5 // Tras el envío de la primera alarma transcurre un
6 // tiempo superior a LIFETIME_RTENTRY, y la ruta al
7 // nodo colector caduca. Al aparecer nuevas alarmas
8 // es necesario iniciar de nuevo el proceso de
9 // Descubrimiento de Ruta.
10 fifo2_out: [0|Q|S|1304427352|D|S|1]                                 // RREQ
11 fifo1_out: [S|P|D|D|1304427353|1|S]                                 // RREP
12 fifo2_out: [D|D|D|S|2|1304427389|W|41.2061|1.7300|87]      // ALARM2
13 fifo1_out: [S|A|S|2]                                              // ACK2
14 fifo2_out: [D|D|D|S|3|1304427393|W|41.2061|1.7300|87]      // ALARM3
15 fifo1_out: [S|A|S|3]                                              // ACK3

```

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304427340	W	41.2061	1.7300	87
2	1304427389	W	41.2061	1.7300	87
3	1304427393	W	41.2061	1.7300	87

Si tras haber utilizado una ruta descubierta transcurre un tiempo sin utilizarla superior a `LIFETIME_RTENTRY`, esta caduca y deja de ser válida. Por ello, si en algún momento se requiere enviar algo al mismo destinatario será necesario volver a realizar el proceso de Descubrimiento de Ruta. Lo ideal será ajustar el valor de `LIFETIME_RTENTRY` en función de la frecuencia de cambio de posición de los nodos en la red. Si hay mucha movilidad interesa que sea un valor pequeño, mientras que si hay poca, o es prácticamente estática, se pueden fijar valores más altos.

8. **Reseteo del nodo D y recepción de alarmas.** El nodo S tiene tres alarmas pendientes de enviar.

```

1 fifo2_out: [0|Q|S|1304430565|D|S|1]                                // RREQ
2 fifo1_out: [S|P|D|D|1304430566|1|S]                                // RREP

```

```

3   fifo2_out: [D|D|D|S|1|1304430528|W|41.2061|1.7300|87]    // ALARM1
4   fifo1_out: [S|A|S|1]                                         // ACK1
5   // En este punto se resetea el nodo D, perdiendo la
6   // información de su tabla de encaminamiento entre
7   // entre otras. El nodo S continúa enviando sus alarmas
8   fifo2_out: [D|D|D|S|2|1304430534|W|41.2061|1.7300|87]    // ALARM2
9   // Al reiniciarse, el nodo D recibe la segunda alarma,
10  // pero su tabla de encaminamiento está vacía y para
11  // poder mandar su acuse de recibo debe encontrar una
12  // ruta hacia el nodo S, por ello esta vez es el nodo D
13  // el que inicial el proceso de Descubrimiento de Ruta
14  fifo1_out: [0|Q|D|1304430573|S|D|1]                           // RREQ
15  fifo2_out: [D|P|S|S|1304430574|1|D]                         // RREP
16  // Mientras que se completa el Descubrimiento de Ruta
17  // transcurre ALARM_TIMEOUT y el nodo S vuelve a
18  // enviar su segunda alarma
19  fifo2_out: [D|D|D|S|2|1304430534|W|41.2061|1.7300|87]    // ALARM2
20  // Esta vez el nodo D ya tiene una ruta válida en su
21  // tabla de encaminamiento y puede mandar su ACK
22  // correspondiente al nodo S
23  fifo1_out: [S|A|S|2]                                         // ACK2
24  fifo2_out: [D|D|D|S|3|1304430539|W|41.2061|1.7300|87]    // ALARM3
25  fifo1_out: [S|A|S|3]                                         // ACK3

```

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304430528	W	41.2061	1.7300	87
2	1304430534	W	41.2061	1.7300	87
2	1304430534	W	41.2061	1.7300	87
3	1304430539	W	41.2061	1.7300	87

Como se comentaba en el punto 6, cuando un nodo se resetea pierde, entre otra información, su tabla de encaminamiento. En este caso, esto le ocurre al nodo D. Sucede que tras reiniciarse, recibe una alarma del nodo S, y debe arrancar el proceso de Descubrimiento de Ruta para poder contestarle con un ACK. Cuando esta operación termina, envía el correspondiente acuse de recibo y todo sigue funcionando con normalidad. Si el tiempo que tarda en descubrirse la nueva ruta es superior a `ALARM_TIMEOUT`, como ocurre en este caso, el nodo S volverá a enviar su alarma, y por eso vemos la número dos repetida en el fichero `alarmReg.dat`.

#### 5.4.3. Escenario 2

El segundo escenario a simular consta de tres nodos alineados, dos de ellos generadores de alarmas, nodo S y nodo A, y el otro colector, nodo D. Su configuración se puede apreciar en la Figura 5.15. Ya que el nodo S no tiene conexión directa con el nodo D, cuando estos requieran comunicarse dependerán del nodo A, que actuará de puente entre ambos.

A continuación, se detallan las diferentes situaciones simuladas sobre este escenario y se comentan los resultados obtenidos. Destacar que `fifo3_out` es lo transmitido por el nodo S, `fifo2_out` lo del nodo A y `fifo1_out` lo del nodo D.

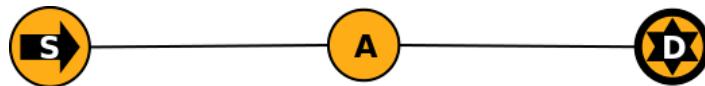


Figura 5.15: Escenario 2 de simulación:  
Enlace bidireccional entre tres nodos alineados.

- Descubrimiento de nodos no vecinos, envío de alarmas y acuses de recibo. Multisalto.** El nodo S tiene dos alarmas pendientes de enviar y el nodo A ninguna.

```

1 // El nodo S envía en modo broadcast un RREQ buscando una
2 // ruta hacia el nodo D
3 fifo3_out: [0|Q|S|1304433773|D|S|1] // RREQ
4 // El nodo A recibe el RREQ del nodo S, guarda una ruta
5 // hacia él en su tabla de encaminamiento y vuelve a
6 // enviar en modo broadcast el mensaje RREQ con sus
7 // campos correspondientes actualizados
8 fifo2_out: [0|Q|S|1304433773|D|A|2] // RREQ
9 // Cuando el nodo D recibe el RREQ, guarda la ruta hacia
10 // el nodo S a través del A, y responde con un RREP unicast
11 fifo1_out: [A|P|D|D|1304433774|1|S] // RREP
12 // El nodo A recibe el RREP del nodo D, guarda una ruta
13 // hacia él en su tabla de encaminamiento y vuelve a
14 // enviar en modo unicast el mensaje RREP con sus
15 // campos correspondientes actualizados al nodo S
16 fifo2_out: [S|P|A|D|1304433774|2|S] // RREP
17 // Tras recibir el nodo S el RREP, conoce una ruta hacia
18 // el nodo D y envía la primera alarma
19 fifo3_out: [A|D|D|S|1|1304433727|W|41.2061|1.7300|87] // ALARM1
20 // El nodo A recibe la alarma 1 del nodo S y la reenvía
21 // según su tabla de encaminamiento al nodo D
22 fifo2_out: [D|D|D|S|1|1304433727|W|41.2061|1.7300|87] // ALARM1
23 // Cuando el nodo D recibe la alarma envía un acuse de
24 // recibo hacia el nodo S, a través del A
25 fifo1_out: [A|A|S|1] // ACK1
26 // El nodo A recibe el ACK1 del nodo D y lo reenvía
27 // según su tabla de encaminamiento al nodo S
28 fifo2_out: [S|A|S|1] // ACK1
29 // Sólo cuando el nodo S recibe el acuse de recibo de la
30 // alarma anterior, procede a enviar la siguiente
31 fifo3_out: [A|D|D|S|2|1304433732|W|41.2061|1.7300|87] // ALARM2
32 fifo2_out: [D|D|D|S|2|1304433732|W|41.2061|1.7300|87] // ALARM2
33 fifo1_out: [A|A|S|2] // ACK2
34 fifo2_out: [S|A|S|2] // ACK2
35 // El nodo S recibe el acuse de recibo y como no tiene más
36 // alarmas pendientes, por el momento no envía nada más.

```

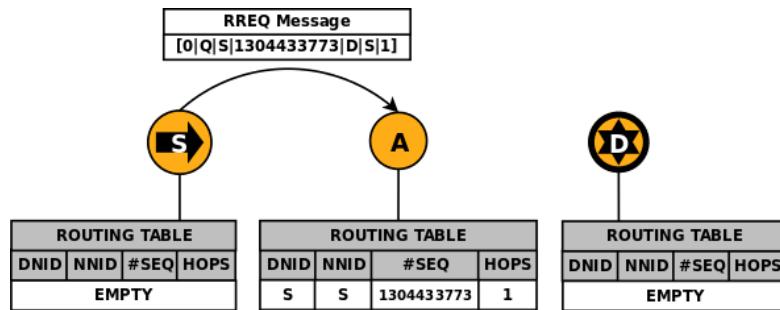
Fichero alarmReg.dat generado por el nodo D:

El sistema es capaz de descubrir rutas hacia nodos no vecinos y, de igual manera, hacerles llegar alarmas y acuses de recibo. Esta característica se conoce con el nombre de multisalto. El parámetro MAX\_NUM\_HOPS del fichero de configuración

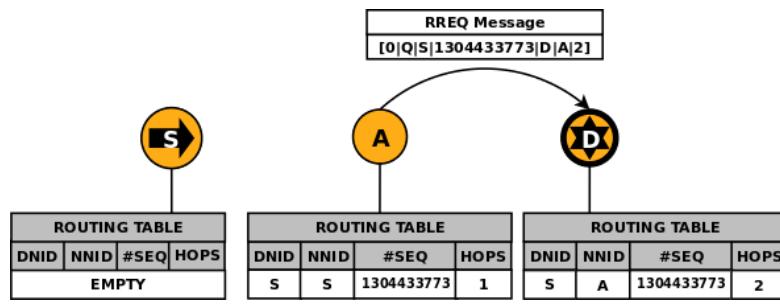
alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304433727	W	41.2061	1.7300	87
2	1304433732	W	41.2061	1.7300	87

`node.conf` determinará el número de saltos que puede haber entre dos nodos que se quieren comunicar.

En las Figuras 5.16 y 5.17, se muestra gráficamente la formación de los caminos de vuelta e ida indicando, en cada caso, el contenido de la tabla de encaminamiento de cada nodo.



(a) El nodo A recibe el RREQ que el nodo S envía en modo broadcast.



(b) El nodo D recibe el RREQ que el nodo A reenvía en modo broadcast.

Figura 5.16: Formación del camino de vuelta del nodo D al S.

## 2. Descarte de paquetes no dirigidos al propio nodo.

Por clarificar la Figura 5.16b se ha representado el mensaje RREQ que recibe el nodo D del nodo A, pero en realidad, al retransmitirse ese mensaje con una antena omnidireccional también le llega al nodo S, que fué el que inició el Descubrimiento de Ruta. Este nodo, en primer lugar, no descarta el mensaje ya que el campo NNID indica modo broadcast y, por lo tanto, debe continuar procesándolo. Es más adelante, al comprobar que el campo SNID del mensaje corresponde con él mismo, cuando reconoce que ese mensaje fué generado por él y lo descarta. Por pantalla, el nodo S devuelve los siguientes mensajes:

```
RREQ message sent (24 bytes): [0|Q|S|1304433773|D|S|1]
Timeout!
Serial data received (25 bytes): [0|Q|S|1304433773|D|A|2]
RREQ message received!
This message is not for me.
Serial data received (25 bytes): [S|P|A|D|1304433774|2|S]
```

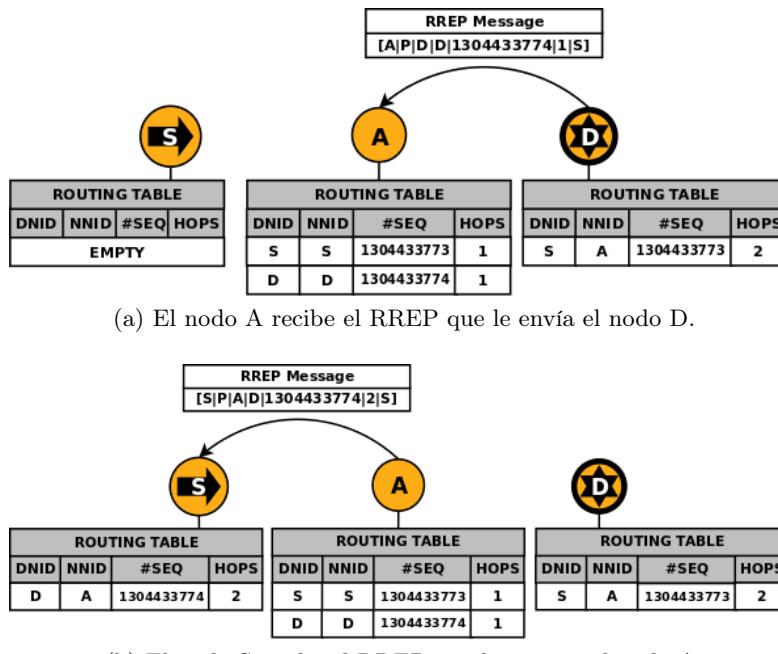


Figura 5.17: Formación del camino de ida del nodos S al D.

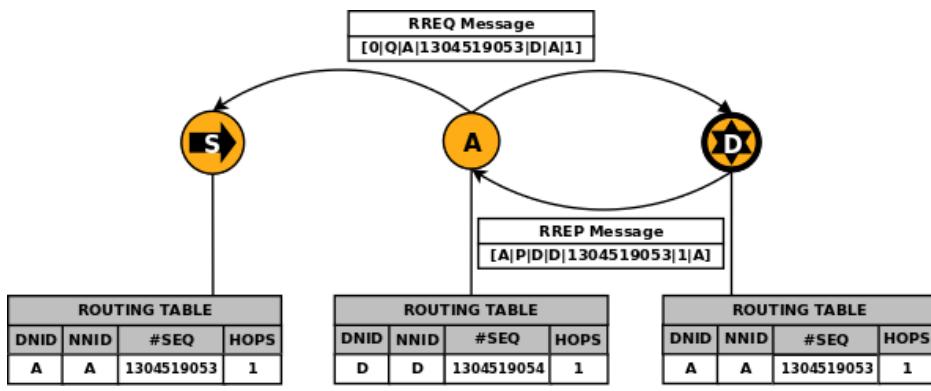
RREP message received!

Cuando un nodo recibe un mensaje que no se ha transmitido en broadcast y él no es el NNID, lo descarta de inmediato. Esto sucede por ejemplo cuando el nodo S envía su alarma hacia el nodo D y recibe la retransmisión que hace de esta el nodo A. Como las alarmas se envian de modo unicast, el nodo S descarta de inmediato ese mensaje y continua esperando su correspondiente ACK. Esta es la salida por pantalla del nodo S:

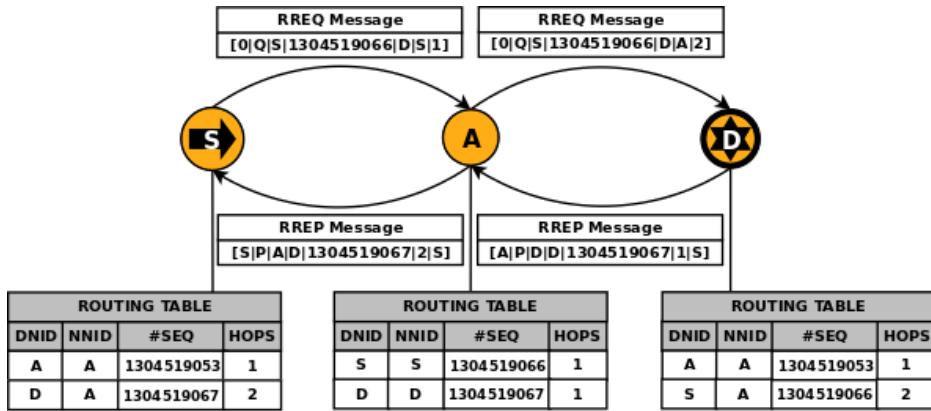
```
Alarm message sent (42 bytes): [A|D|D|S|1|1304433727|W|41.2061|1.7300|87]
Timeout!
Serial data received (43 bytes): [D|D|D|S|1|1304433727|W|41.2061|1.7300|87]
This message is not for me.
Serial data received (10 bytes): [S|A|S|1]
ACK message received!
```

### 3. Actualización de tablas de encaminamiento.

Inicialmente el nodo A dispone de varias alarmas que debe hacer llegar al nodo D. Por ello, realiza un Descubrimiento de Ruta hacia dicho nodo, como se puede observar en la Figura 5.18a. En ella, se muestran además las tablas de encaminamiento de todos los nodos al finalizar el proceso. A continuación, el nodo S empieza a generar alarmas y, como no tiene una ruta hacia el nodo colector, debe realizar otro Descubrimiento de Ruta, que se representa en la Figura 5.18b. Se puede observar como en las tablas de los nodos S y D se incorpora una entrada con la nueva ruta descubierta. La tabla del nodo A actualiza la entrada que ya tenía por una ruta más actual, es decir, con un número de secuencia mayor, y



(a) El nodo A realiza un Descubrimiento de Ruta hacia el nodo D.



(b) El nodo S realiza un Descubrimiento de Ruta hacia el nodo D.

Figura 5.18: Actualización de tablas de encaminamiento.

añade una nueva ruta hacia el nodo S. De esta manera, quedan todas actualizadas a la nueva situación en la que ambos nodos pueden enviar sus alarmas al nodo colector.

- 4. Envío de alarmas desde dos nodos.** En la simulación, el nodo A envía siete alarmas y el nodo S cinco. En este caso se han modificado los campos `alarmType` de las alarmas poniendo en su lugar el identificador del nodo que las envía. De esta manera, al concluir la simulación, en el fichero `alarmReg.dat` del nodo D se puede observar en qué orden llegaron y si se entregaron todas, como en este caso.

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304441541	A	41.2061	1.7300	87
2	1304441548	A	41.2061	1.7300	87
1	1304441543	S	41.2061	1.7300	87
3	1304441554	A	41.2061	1.7300	87
2	1304441549	S	41.2061	1.7300	87
4	1304441561	A	41.2061	1.7300	87
5	1304441566	A	41.2061	1.7300	87
3	1304441552	S	41.2061	1.7300	87
6	1304441570	A	41.2061	1.7300	87
7	1304441577	A	41.2061	1.7300	87
4	1304441555	S	41.2061	1.7300	87
5	1304441559	S	41.2061	1.7300	87

Como se puede ver, todas las alarmas llegaron al nodo colector correctamente. Las del nodo A se entregaron antes a pesar de ser más, ya que la ruta que siguieron no era multisalto.

- 5. Envío de alarmas desde dos nodos con pérdida temporal de la comunicación.** El nodo A entrega cinco alarmas y el nodo S seis. Durante el proceso se pierde temporalmente la comunicación entre los nodos generadores.

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304511821	A	41.2061	1.7300	87
2	1304511827	A	41.2061	1.7300	87
3	1304511830	A	41.2061	1.7300	87
1	1304511825	S	41.2061	1.7300	87
4	1304511821	A	41.2061	1.7300	87
1	1304511825	S	41.2061	1.7300	87
5	1304511856	A	41.2061	1.7300	87
2	1304511830	S	41.2061	1.7300	87
3	1304511837	S	41.2061	1.7300	87
4	1304511843	S	41.2061	1.7300	87
5	1304511850	S	41.2061	1.7300	87
6	1304511856	S	41.2061	1.7300	87

Llegan todas las alarmas correctamente al nodo colector. Lo único destacable es que la alarma 1 del nodo S llega repetida ya que es entregada correctamente la primera vez pero; al perderse temporalmente la comunicación entre el nodo A y el S, éste no recibe el ACK correspondiente, por lo que tras `ALARM_TIMEOUT` la reenvía.

- 6. Solicitud de posición de los nodos.** Durante el envío de los mensajes PREQ y PREP no se están enviando alarmas.

```

1 // El nodo D realiza una solicitud de posición al
2 // resto de nodos de la red
3 fifo1_out: [0|W|D|1304516600|D|1]
4 // El nodo A recibe la solicitud, la responde con
5 // mensaje PREP y reenvía esa solicitud nuevamente
6 // para propagarla por la red
7 fifo2_out: [D|C|D|1304516621|A|+90.1234|-127.4321]
8 fifo2_out: [0|W|D|1304516600|A|2]
9 // El nodo S recibe la solicitud que reenvió el nodo
10 // A, la responde con un mensaje PREP y la reenvía
11 // nuevamente para propagarla por la red
12 fifo3_out: [A|C|D|1304516622|S|+90.1234|-127.4321]
13 fifo3_out: [0|W|D|1304516600|S|3]
14 // Cuando el nodo A recibe el PREP del nodo S, lo
15 // envía según su tabla de encaminamiento hacia el
16 // nodo D
17 fifo2_out: [D|C|D|1304516622|S|+90.1234|-127.4321]

```

Fichero `positionReg.dat` generado por el nodo D:

timeStamp	NID	latitude	longitude
1304516621	A	+90.1234	-127.4321
1304516622	S	+90.1234	-127.4321

Observando los timestamp correspondientes a los nodos A y S, vemos que la respuesta de éste último se generó un segundo después que la del nodo A. Por lo tanto, en un caso real esa diferencia de tiempo será igual o mayor a ese valor.

- Envío conjunto de alarmas, **PREQ** y **PREP**. El nodo A envía cuatro alarmas, el S otras cuatro y el nodo D realiza dos solicitudes de posición.

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304516437	S	41.2061	1.7300	87
1	1304516442	A	41.2061	1.7300	87
2	1304516449	A	41.2061	1.7300	87
2	1304516443	S	41.2061	1.7300	87
3	1304516453	A	41.2061	1.7300	87
4	1304516456	A	41.2061	1.7300	87
3	1304516449	S	41.2061	1.7300	87
4	1304516454	S	41.2061	1.7300	87

Fichero `positionReg.dat` generado por el nodo D:

timeStamp	NID	latitude	longitude
1304516448	A	+90.1234	-127.4321
1304516452	S	+90.1234	-127.4321
1304516466	A	+90.1234	-127.4321
1304516469	S	+90.1234	-127.4321

Tanto el envío de alarmas como las solicitudes de posición se realizan satisfactoriamente. En este caso, al haber más tráfico en la red, la diferencia de tiempo entre los timestamp de los mensajes PREP es mayor que en el caso anterior.

#### 5.4.4. Escenario 3

El tercer escenario a simular consta de cinco nodos. Como se observa en la Figura 5.19, están configurados de modo que entre el nodo S y el nodo D existen dos posibles caminos. Uno de ellos es idéntico al del escenario dos, y el otro depende de los nodos B y C, por consiguiente, cuenta con tres saltos, en este caso el máximo permitido para una ruta. De esta manera, se rompe la dependencia del nodo A que tenían los nodos S y D para comunicarse, ya que en caso de fallo existiría una ruta alternativa.

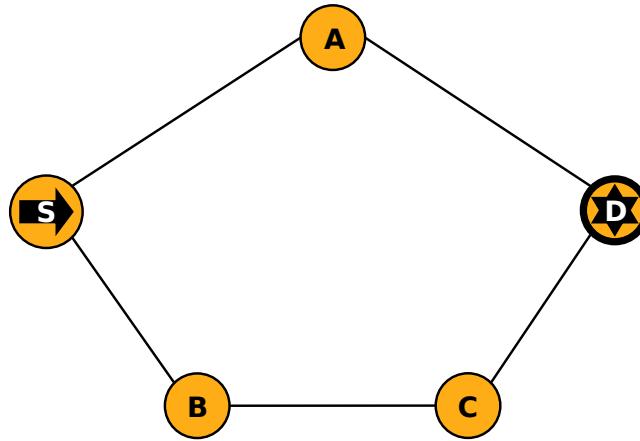


Figura 5.19: Escenario 3 de simulación: Varias rutas distintas entre dos nodos.

A continuación, se detallan las diferentes situaciones simuladas sobre este escenario y se comentan los resultados obtenidos.

##### 1. Descubrimiento de la ruta más corta entre los nodos S y D.

Partiendo de que todos los nodos tienen sus tablas de encaminamiento vacías, al realizar el S un Descubrimiento de Ruta hacia el D, el resultado es el que se observa en la Figura 5.20. El nodo D, en condiciones ideales, recibirá un RREQ reenviado por el A y otro por el C. Independientemente de cuál de los dos le llegue primero, la ruta elegida finalmente será la que cuente con el menor número de saltos, en este caso a través del nodo A. Esto es así ya que los dos mensajes RREQ llevarán el mismo número de secuencia, por lo cual, pertenecerán al mismo Descubrimiento de Ruta. En caso de que tuvieran un número de secuencia distinto se daría prioridad a la ruta más actual, aunque tuviese un mayor número de saltos. En el caso simulado se extrae que el RREQ llegó primero a través del nodo A. Por eso, los nodos B y C únicamente tienen rutas hacia el nodo S, y no hacia el D, ya que estas se habrían formado al propagar un mensaje RREP.

##### 2. Redescubrimiento de ruta. Primera prueba.

Inicialmente los cinco nodos de la red están activos. El nodo S tiene cuatro alarmas pendientes de entregar. Por ello, realiza un Descubrimiento de Ruta que, como se explicó en el punto anterior, al finalizar deja su tabla de encaminamiento como en la Figura 5.21a. Una vez dispone de una ruta válida comienza a enviar sus alarmas. Tras las dos primeras, el nodo A falla. El S agota los reintentos de envío de la alarma 3 y, al no tener respuesta, la descarta. Como todavía tiene una cuarta alarma que entregar, relanza un Descubrimiento de Ruta obteniendo

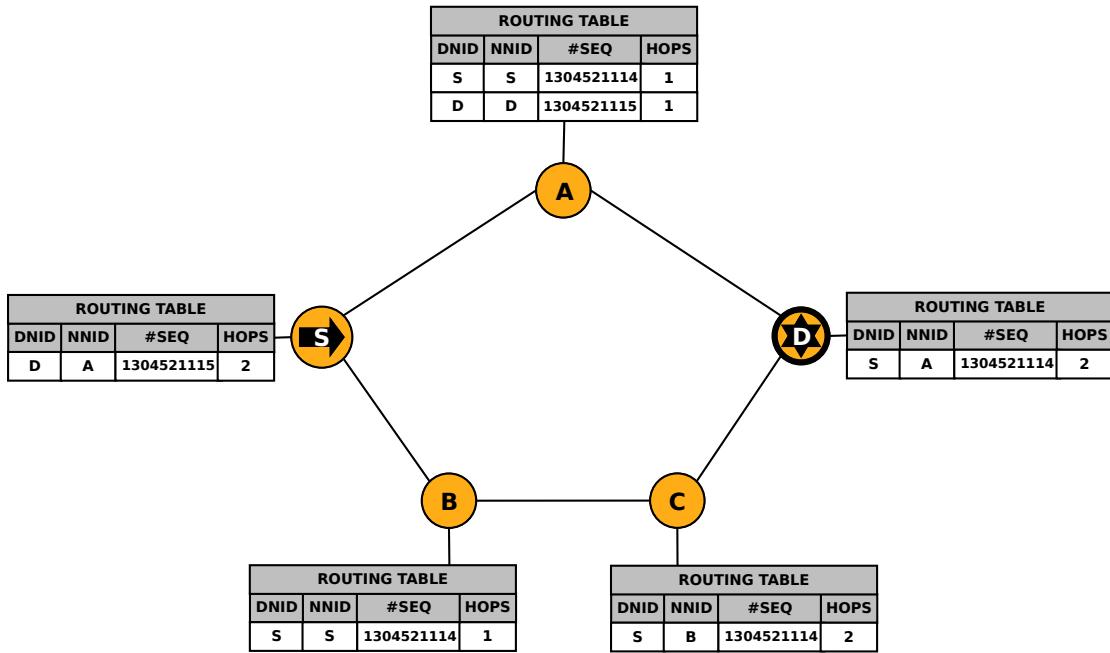


Figura 5.20: Descubrimiento de la ruta más corta entre los nodos S y D.

una nueva ruta a través de los nodos B y C, como se observa en la Figura 5.21b. A través de ella, logra enviar su última alarma.

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
D	A	1304523718	2

(a) Tabla de encaminamiento del nodo S tras el primer Descubrimiento de Ruta.

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
D	B	1304523719	3

(b) Tabla de encaminamiento del nodo S tras el segundo Descubrimiento de Ruta.

Figura 5.21: Evolución de las tablas de encaminamiento del nodo S.

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304523038	S	41.2061	1.7300	87
2	1304523044	S	41.2061	1.7300	87
4	1304523052	S	41.2061	1.7300	87

Como se puede ver, la caída por un periodo prolongado de tiempo del nodo A supuso la pérdida de la tercera alarma.

### 3. Redescubrimiento de ruta. Segunda prueba.

Esta prueba es una continuación de la anterior. Inicialmente el nodo S tiene ocho alarmas que entregar. Después del primer Descubrimiento de Ruta (ver Figura 5.22a) se envían las dos primeras alarmas. El nodo A deja de operar en la red, y trás perderse la alarma 3, el S vuelve a buscar un camino para enviar sus alarmas (ver Figura 5.22b). En este momento, el nodo A vuelve a estar activo, pero mientras que la nueva ruta no caduque o de problemas no se cambiaría, aunque esté operativa una más corta. Tras enviar la alarma 4, el nodo C deja de operar. Con la quinta alarma ocurre lo mismo que con la tercera y a continuación se realiza un nuevo Descubrimiento de Ruta (ver Figura 5.22c). Por este último camino se logran entregar las últimas tres alarmas.

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
D	A	1304527513	2

(a) Tabla de encaminamiento del nodo S tras el primer Descubrimiento de Ruta.

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
D	B	1304527514	3

(b) Tabla de encaminamiento del nodo S tras el segundo Descubrimiento de Ruta.

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
D	A	1304527515	2

(c) Tabla de encaminamiento del nodo S tras el tercer Descubrimiento de Ruta.

Figura 5.22: Evolución de las tablas de encaminamiento del nodo S.

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304524407	S	41.2061	1.7300	87
2	1304524411	S	41.2061	1.7300	87
4	1304524418	S	41.2061	1.7300	87
6	1304524428	S	41.2061	1.7300	87
7	1304524434	S	41.2061	1.7300	87
8	1304524441	S	41.2061	1.7300	87

Tras esta prueba y la anterior, se puede concluir que ante caídas prolongadas de nodos que forman parte de una ruta activa se perderá siempre una alarma.

#### 4. Envío de alarmas desde cuatro nodos. Influencia del parámetro ALARM\_TIMEOUT en el tráfico.

Cada nodo de la red tiene dos alarmas para entregar al nodo colector. Tras los Descubrimientos de Ruta, la tabla de encaminamiento del nodo D se muestra en la Figura 5.23.

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
C	C	1304528418	1
A	A	1304528420	1
S	A	1304528417	2
B	C	1304528420	2

Figura 5.23: Tabla de encaminamiento del nodo D.

En primer lugar, el parámetro de configuración `ALARM_TIMEOUT` se fija a 8 segundos y el fichero `alarmReg.dat` generado tras la simulación por el nodo D es:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304528134	C	41.2061	1.7300	87
1	1304528134	A	41.2061	1.7300	87
2	1304528137	C	41.2061	1.7300	87
2	1304528137	A	41.2061	1.7300	87
1	1304528134	S	41.2061	1.7300	87
1	1304528134	B	41.2061	1.7300	87
1	1304528134	S	41.2061	1.7300	87
2	1304528137	S	41.2061	1.7300	87
2	1304528137	B	41.2061	1.7300	87

En este caso, la primera alarma del nodo S se envía y recibe dos veces. Esto se debe a que la primera vez su ACK tarda en llegar más de `ALARM_TIMEOUT` por el tráfico de la red, y el nodo reintenta el envío. Esto solo ocurre con la primera ya que es el momento en el que más tráfico hay en la red.

A continuación, se repite la prueba pero esta vez con el parámetro `ALARM_TIMEOUT` a 12 segundos. El fichero `alarmReg.dat` generado por el nodo D es el siguiente:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304529073	C	41.2061	1.7300	87
1	1304529073	A	41.2061	1.7300	87
2	1304529080	C	41.2061	1.7300	87
2	1304529080	A	41.2061	1.7300	87
1	1304529073	S	41.2061	1.7300	87
1	1304529073	B	41.2061	1.7300	87
2	1304529080	S	41.2061	1.7300	87
2	1304529080	B	41.2061	1.7300	87

En este segundo caso, ninguna alarma llega repetida. Se observa que las alarmas de los **nodos A** y **C** llegan antes que las de **S** y **B**. Esto se debe a que los primeros únicamente tiene un salto de distancia con el nodo colector, mientras que los otros tienen dos. Por último, resaltar que el parámetro **ALARM\_TIMEOUT** deberá ajustarse correctamente según la aplicación para optimizar la utilización de recursos de la red.

##### 5. Envío de alarmas desde tres nodos en línea.

Realizando una pequeña modificación del fichero **bash.nodes** se configura la red de modo que el **nodo A** no puede comunicar con ningún otro y el resto se mantienen como antes. De este modo, se logra tener tres nodos generadores de alarmas en línea que envian a un nodo colector. Como se puede observar en la Figura 5.24, en la tabla de encaminamiento del **nodo D** se vé que la distancia a los **nodos C, B** y **S** es de uno, dos y tres saltos, respectivamente. Cada uno de estos nodos tiene dos alarmas para enviar.

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
C	C	1304529688	1
B	C	1304529688	2
S	C	1304529688	3

Figura 5.24: Tabla de encaminamiento del nodo D.

Igual que en el caso anterior, inicialmente se realiza la simulación con el parámetro **ALARM\_TIMEOUT** a 8 segundos. El fichero **alarmReg.dat** generado por el **nodo D** es el siguiente:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304529073	C	41.2061	1.7300	87
2	1304529080	C	41.2061	1.7300	87
1	1304529073	B	41.2061	1.7300	87
1	1304529414	S	41.2061	1.7300	87
2	1304529080	B	41.2061	1.7300	87
1	1304529414	S	41.2061	1.7300	87
2	1304529420	S	41.2061	1.7300	87

Nuevamente, y por los mismos motivos, observamos que la primera alarma del **nodo S** se vuelve a entregar duplicada.

Al repetir la simulación modificando **ALARM\_TIMEOUT** a 12 segundos, los resultados son:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304530117	C	41.2061	1.7300	87
2	1304530120	C	41.2061	1.7300	87
1	1304530117	B	41.2061	1.7300	87
1	1304530117	S	41.2061	1.7300	87
2	1304530120	B	41.2061	1.7300	87
2	1304530120	S	41.2061	1.7300	87

Se concluye que en redes con muchos nodos, donde puede haber varias alarmas que se entreguen a la vez, el parámetro `ALARM_TIMEOUT` deberá ser seleccionado a conciencia para evitar reenvíos innecesarios y aumentar de esa manera el tráfico de la misma.

#### 5.4.5. Escenario 4

El cuarto, y último, escenario a simular consta de cinco nodos. Como se observa en la Figura 5.25, se representa una red dinámica en la que inicialmente el nodo D únicamente puede comunicarse con el C. A continuación, tras un desplazamiento, esta conexión se perderá y el nodo D establecerá una nueva con el B. Este hecho reducirá en un salto la distancia entre S y D. Para simular esta situación, se han implementado los ficheros `bash1.nodes` y `bash2.nodes`, correspondientes relativamente a las situaciones descritas.

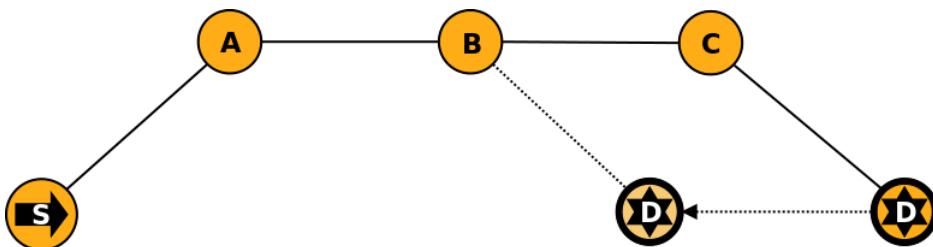


Figura 5.25: Escenario 4 de simulación: Red dinámica.

A continuación, se detallan las diferentes situaciones simuladas sobre este escenario y se comentan los resultados obtenidos.

##### 1. Límite de saltos de una ruta.

En la configuración de todos los nodos de la red se fija el número máximo de saltos en una ruta, `MAX_NUM_HOPS`, a tres. El nodo S tiene alarmas que enviar al D y para ello debe realizar un Descubrimiento de Ruta. Con la configuración inicial nunca encontrará dicho camino, ya que la distancia entre estos nodos es de cuatro saltos. En la simulación vemos lo siguiente:

```

1 fifo3_out: [0|Q|S|1304601056|D|S|1]      // RREQ1
2 // El nodo A lo recibe, ve que el campo 'hops' es menor
3 // que MAX_NUM_HOPS y lo reenvía
4 fifo2_out: [0|Q|S|1304601056|D|A|2]      // RREQ1
5 // El nodo B lo recibe, ve que el campo 'hops' es menor
6 // que MAX_NUM_HOPS y lo reenvía
7 fifo4_out: [0|Q|S|1304601056|D|B|3]      // RREQ1
8 // El nodo C lo recibe, ve que el campo 'hops' es igual
9 // que MAX_NUM_HOPS y no lo reenvía.
10 // Tras RREQ_TIMEOUT el nodo S repite la operación.
  
```

```

11 fifo3_out: [0|Q|S|1304601057|D|S|1]      // RREQ2
12 fifo2_out: [0|Q|S|1304601057|D|A|2]      // RREQ2
13 fifo4_out: [0|Q|S|1304601057|D|B|3]      // RREQ2

```

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
S	B	1304601057	3

Figura 5.26: Tabla de encaminamiento del nodo C.

Tras el mensaje RREQ del nodo S, los tres siguientes nodos crean una ruta hacia él, pero el tercero de ellos, en este caso el nodo C, compara el campo hops del mensaje RREQ con el parámetro MAX\_NUM\_HOPS de su configuración y, al ver que son iguales, ya no reenvía nuevamente el RREQ. Por ello, el nodo D nunca recibe la solicitud de ruta del S y la alarma o alarmas de éste no podrán ser entregadas. Como ya se comentó anteriormente, la utilización de una restricción de saltos para una ruta permite controlar la dispersión de los mensajes en la red.

## 2. Dinamicidad de la red.

El nodo D que, anteriormente, no era alcanzable desde el nodo S se mueve y pasa a serlo.

```

1 // Se comienza con la primera configuración correspondiente
2 // a 'bash1.nodes'
3 $ ./bash_router.sh bash1.nodes
4 fifo3_out: [0|Q|S|1304601583|D|S|1]      // RREQ1 (NODO S)
5 fifo2_out: [0|Q|S|1304601583|D|A|2]      // RREQ1 (NODO A)
6 fifo4_out: [0|Q|S|1304601583|D|B|3]      // RREQ1 (NODO B)
7 fifo3_out: [0|Q|S|1304601584|D|S|1]      // RREQ2 (NODO S)
8 fifo2_out: [0|Q|S|1304601584|D|A|2]      // RREQ2 (NODO A)
9 fifo4_out: [0|Q|S|1304601584|D|B|3]      // RREQ2 (NODO B)
10 // Se cambia a la segunda configuración correspondiente
11 // a 'bash2.nodes'
12 $ ./bash_router.sh bash2.nodes
13 fifo3_out: [0|Q|S|1304601585|D|S|1]      // RREQ3 (NODO S)
14 fifo2_out: [0|Q|S|1304601585|D|A|2]      // RREQ3 (NODO A)
15 fifo4_out: [0|Q|S|1304601585|D|B|3]      // RREQ3 (NODO B)
16 // Cuando el nodo B reenvía el RREQ el nodo D lo recibe
17 // por fin, y responde con el correspondiente RREP
18 fifo1_out: [B|P|D|1304601586|1|S]      // RREP (NODO D)
19 fifo4_out: [A|P|B|D|1304601586|2|S]      // RREP (NODO B)
20 fifo2_out: [S|P|A|D|1304601586|3|S]      // RREP (NODO A)
21 // El nodo S recibe el RREP del nodo A y manda su alarma
22 fifo3_out: [A|D|D|S|1|1304532488|S|41.2061|1.7300|87] // ALARM1
23 // El nodo A reenvía la alarma 1 del nodo S
24 fifo2_out: [B|D|D|S|1|1304532488|S|41.2061|1.7300|87] // ALARM1
25 // El nodo B reenvía la alarma 1 del nodo S
26 fifo4_out: [D|D|D|S|1|1304532488|S|41.2061|1.7300|87] // ALARM1
27 // Cuando el nodo B reenvía la alarma 1 del nodo S, el
28 // nodo D la recibe y responde con el correspondiente
29 // acuse de recibo (ACK1)
30 fifo1_out: [B|A|S|1]      // ACK1 (NODO D)

```

```

31 |     fifo4_out: [A|A|S|1]      // ACK1 (NODO B)
32 |     fifo2_out: [S|A|S|1]      // ACK1 (NODO A)
33 | // El nodo S recibe el ACK de su primera alarma y si tuviera
34 | // más alarmas las seguiría enviando

```

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
S	B	1304601585	3

Figura 5.27: Tabla de encaminamiento del nodo D.

Cuando el nodo D cambia de posición recibe el RREQ del nodo S y responde con el correspondiente RREP, formando una ruta entre ambos. Una vez que el RREP llega a S, éste comienza a enviar sus alarmas que serán respondidas por D con acuses de recibo.

El protocolo de encaminamiento AODV-LAB soporta y detecta cambios en la topología de la red, de modo que, si existe algún camino posible entre dos nodos para comunicarse se descubrirá y podrá utilizarse.

### 3. Envío de alarmas desde múltiples nodos.

Utilizando la configuración definida por `bash2.nodes`, es decir, la que resulta tras el movimiento del nodo D, se envían dos alarmas desde cada nodo de la red. A continuación, se muestran la tabla de encaminamiento y el fichero `alarmReg.dat` generado por D.

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1304602164	B	41.2061	1.7300	87
2	1304602169	B	41.2061	1.7300	87
1	1304602164	A	41.2061	1.7300	87
1	1304602164	S	41.2061	1.7300	87
1	1304602164	C	41.2061	1.7300	87
2	1304602169	A	41.2061	1.7300	87
2	1304602169	C	41.2061	1.7300	87
2	1304602169	S	41.2061	1.7300	87

ROUTING TABLE			
DNID	NNID	#SEQ	HOPS
B	B	1304602277	1
A	B	1304602278	2
C	B	1304602278	2
S	B	1304602277	3

Figura 5.28: Tabla de encaminamiento del nodo D.

Al igual que sucedió en el escenario 3, se vuelve a observar que los nodos más próximos al colector entregan sus alarmas ántes que los más alejados.

#### **5.4.6. Conclusiones**

Como se comentaba al inicio de esta sección, el objetivo que se perseguía era constatar las principales funcionalidades deseadas, tanto del protocolo de encaminamiento AODV-LAB, como del módulo de entrega remota de datos del sistema LIDO DCL. Además de ello, la simulación también ha servido para verificar el comportamiento deseado en ciertas situaciones y, finalmente, lograr una mejor comprensión del sistema.

Sobre sus funcionalidades se puede afirmar que éste soporta:

- Descubrimiento de rutas hacia nodos vecinos.
- Descubrimiento de la ruta más corta entre dos nodos no vecinos.
- Redescubrimiento de una ruta alternativa entre dos nodos en caso de fallo prolongado de la actual. Este hecho, se vió que produce siempre la pérdida de una alarma en caso de que se estén enviando.
- Envío de alarmas y acuses de recibo entre dos nodos, o simultáneamente desde varios a un nodo colector. Capacidad de multisalto.
- Pérdidas temporales de la comunicación entre nodos sin extravío de datos. Esto es posible gracias a la utilización de acuses de recibo y a la posibilidad de reintentar el envío de alarmas tantas veces como se indique en **ALARM\_RETRIES**.
- Envío de solicitudes y respuestas de posición sin necesidad de realizar un Descubrimiento de Ruta.
- Envío conjunto de alarmas y datos de posición entre varios nodos y un colector.
- Limitación del número máximo de saltos en una ruta para controlar la dispersión de los mensajes en la red.
- Dinamicidad de la red, es decir, detecta cambios en la topología y responde en consecuencia para mantener su operatividad. Caducidad de las entradas en la tabla de encaminamiento.
- Reseteo de los nodos de la red. Éste hecho puede producir que se entreguen alarmas repetidas pero, gracias a que el número de secuencia se inicializa con la función `time()`, al reiniciarse el nodo puede comenzar a operar sin ningún problema en la red.

Además, se ha verificado el comportamiento deseado frente a la siguientes situaciones:

- Agotamiento de los reintentos de envío de la última alarma del fichero **alarm.dat**. En tal caso, ésta se descarta y no se realiza ninguna operación más.
- Agotamiento de los reintentos de envío de una alarma del fichero **alarm.dat** que tiene, además de ésta, otras pendientes de entregar. En este caso, se descarta la que agotó los reintentos y, antes de continuar enviando nuevas alarmas, se realiza un Descubrimiento de Ruta nuevo, ya que el camino anterior no fué válido.
- Un nodo con alarmas pendientes de entregar inicia periódicamente un Descubrimiento de Ruta, hasta que este finalice con éxito y pueda entregar sus alarmas.
- Descarte de mensajes no dirigidos al propio nodo.
- Actualización de las tablas de encaminamiento.
- Influencia del parámetro **ALARM\_TIMEOUT** en el tráfico de la red.

Por todo ello, se considera que el software implementado para este primer prototipo de red ad hoc de datos cumple los objetivos deseados y, por lo tanto, será válido para continuar con las pruebas. Se destaca el enfoque del mismo hacia un uso óptimo de los recursos, característica muy importante en un medio aislado como es el mar. No obstante, si se diera continuidad a este proyecto, mejorar el software sería un objetivo importante a alcanzar, trabajando sobre: los protocolos de encaminamiento, las funcionalidades del módulo de entrega remota de datos, la mejora del código, o todos ellos.



# Capítulo 6

## Prueba final del prototipo de red ad hoc de datos

*Lo importante es no dejar de hacerse preguntas.*

Albert Einstein

**RESUMEN:** Este capítulo describe una serie de pruebas realizadas con el prototipo final de red ad hoc de datos, concretamente utilizando 3 nodos para constituir la red. Los resultados observados muestran algunas situaciones nuevas respecto a las simulaciones. Se confirma el funcionamiento de la red y algunas de las funcionalidades deseadas, pero es necesario profundizar en la realización de este tipo de pruebas, a ser posible en el mar, para: lograr una configuración de los nodos tanto a nivel de hardware como de software más óptima, probar funcionalidades que requieren un mayor número de nodos y detectar posibles fallos del código y corregirlos.

### 6.1. Introducción

Una vez finalizado el desarrollo de las distintas partes del prototipo por separado, se ha procedido a la integración de las mismas según el diseño realizado inicialmente. Este capítulo documenta una serie de pruebas realizadas con tres nodos que emplean el software desarrollado y el hardware seleccionado y probado anteriormente en este proyecto.

Los nodos se han colocado de manera alineada en tierra firme formando un enlace bidireccional en el que, como se muestra en la Figura 6.1, los **nodos S** y **D** requieren del **A** para poderse comunicar entre sí. Los nodos se han formado de la siguiente manera:

- **Nodo D.** Actúa como colector o destino de todas las alarmas. Consiste en un módulo de comunicación (XBee-PRO 868 + Antena + Conversor UART-USB) conectado por el puerto USB a un ordenador de mesa con la distribución Ubuntu 11.04 de GNU/Linux.
- **Nodos A y S.** Actúan como nodos generadores de alarmas, que sería el equivalente a una boyá operando en el mar. Cada uno consiste en un ordenador portátil con Ubuntu 10.04 y un módulo de comunicación conectado por el puerto USB.

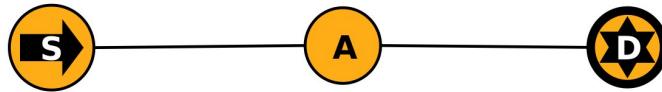


Figura 6.1: Escenario de pruebas: Enlace bidireccional entre tres nodos alineados.

El firmware de todos los XBee-PRO 868 se ha mantenido configurado sin reenvíos a nivel de MAC ( $RR=0$  y  $MT=0$ ). La potencia utilizada ha sido la mínima ( $PL=0$ ) ya que las pruebas se han realizado con los nodos relativamente próximos entre sí. Los parámetros de configuración del software de todos los nodos han sido:

- `SERIAL_NAME = '/dev/ttyUSB0'`
- `SERIAL_BAUD_RATE = 9600`
- `NODE_ID = 'S', 'A' o 'D'`. Cada uno el suyo.
- `SINK_NODE_ID = 'D'`
- `MAX_NUM_HOPS = 3`
- `LIFETIME_RTENTRY = 300`. Valor en segundos.
- `ALARM_RETRIES = 2`
- `ALARM_TIMEOUT = 12`. Valor en segundos.
- `RREQ_TIMEOUT = 12`. Valor en segundos.

## 6.2. Primera prueba

Ha consistido en el envío de 4 alarmas desde en nodo S al D.

Durante su operación cada nodo va imprimiendo mensajes por pantalla indicando los datos que recibe y los que envía. De esta manera es posible, al finalizar cada prueba, observar cómo se ha desarrollado todo. Se han registrado los mensajes del nodo D y a continuación se presentan comentados, permitiendo así seguir y entender lo sucedido en el proceso de entrega de las 4 alarmas.

```

1 // Se recibe un RREQ del nodo S a través del A. Se puede observar que
2 // el número de saltos es 2.
3 Serial data received (24 bytes): [0|Q|S|1310577406|D|A|2]
4 RREQ message received!
5
6 // Se guarda la ruta hacia el nodo S y se responde al RREQ con un
7 // RREP unicast.
8 Unicast RREP message (24 bytes): [A|P|D|D|1310577407|1|S]
9 // Se imprime por pantalla la tabla de rutas y de registro de RREPs.
10 ****
11 *   DNID      NNID      SEQN      HOPS      *
12 *   S          A          1310577406    2          *
13

```

```
14 ****
15 ****
16 ***          RREQ TABLE      ***
17 *-----*
18 *      SNID      SeqNum      *
19 *-----*
20 *      S          1310577406      *
21 ****
22
23 // Se recibe el RREP que el nodo A envía al S, y como es unicast se descarta.
24 Serial data received (24 bytes): [S|P|A|D|1310577407|2|S]
25 This message is not for me.
26
27 // Llega la primera alarma. El nodo S la mandó al A, y éste al D.
28 Serial data received (42 bytes):
29     [D|D|D|S|1|1310564064|S|41.2061|1.7300|87]
30 DATA message received!
31
32 // Se imprime la información de la alarma recibida.
33 Alarm received: alID: 1           timeStamp: 1310564064      alType: S
34             Lat: 41.2061        Lon: 1.7300                  PerConf: 87
35
36 // Se envía el acuse de recibo (ACK) de la primera alarma.
37 Unicast ACK message (9 bytes): [A|A|S|1]
38
39 // Se vuelve a recibir la primera alarma. Como no llegó el ACK reenviado por el nodo A, se sabe que ese mensaje falló entre el D y el A y, por eso, tras ALARM_TIMEOUT el nodo S vuelve a enviar su primera alarma.
40 Serial data received (42 bytes):
41     [D|D|D|S|1|1310564064|S|41.2061|1.7300|87]
42 DATA message received!
43 Alarm received: alID: 1           timeStamp: 1310564064      alType: S
44             Lat: 41.2061        Lon: 1.7300                  PerConf: 87
45
46 // Se envía el ACK de la primera alarma.
47 Unicast ACK message (9 bytes): [A|A|S|1]
48
49 // Llega el ACK reenviado por el nodo A. Como va dirigido a S, el nodo D lo descarta.
50 Serial data received (9 bytes): [S|A|S|1]
51 This message is not for me.
52
53 // Llega la alarma 2 que el nodo S envía al A. Esto significa que por algún motivo los nodos S y D se encuentran uno dentro del área de cobertura del otro pero, como el mensaje es unicast, el nodo D lo descarta.
54 Serial data received (42 bytes):
55     [A|D|D|S|2|1310564070|S|41.2061|1.7300|87]
56 This message is not for me.
57
58 // Ahora se recibe la alarma 2 reenviada por el nodo A. Como D es el destinatario se registra la alarma y se imprime por pantalla.
59 Serial data received (42 bytes):
60     [D|D|D|S|2|1310564070|S|41.2061|1.7300|87]
```

```

61 DATA message received!
62 Alarm received: alID: 2           timeStamp: 1310564070      alType: S
63             Lat: 41.2061          Lon: 1.7300                  PerConf: 87
64
65 //Se envía el ACK de la segunda alarma.
66 Unicast ACK message (9 bytes): [A|A|S|2]
67
68 // Llega el ACK reenviado por el nodo A. Como va dirigido a S, el
   nodo D lo descarta.
69 Serial data received (9 bytes): [S|A|S|2]
70 This message is not for me.
71
72 // Se recibe la alarma 3.
73 Serial data received (42 bytes):
74     [D|D|D|S|3|1310564077|S|41.2061|1.7300|87]
75 DATA message received!
76 Alarm received: alID: 3           timeStamp: 1310564077      alType: S
77             Lat: 41.2061          Lon: 1.7300                  PerConf: 87
78
79 // Envio de ACKs por los nodos D y A.
80 Unicast ACK message (9 bytes): [A|A|S|3]
81 Serial data received (9 bytes): [S|A|S|3]
82 This message is not for me.
83
84 // Se recibe la alarma 4.
85 Serial data received (42 bytes):
86     [D|D|D|S|4|1310564082|S|41.2061|1.7300|87]
87 DATA message received!
88 Alarm received: alID: 4           timeStamp: 1310564082      alType: S
89             Lat: 41.2061          Lon: 1.7300                  PerConf: 87
90
91 // Envío de ACKs por los nodos D y A.
92 Unicast ACK message (9 bytes): [A|A|S|4]
93 Serial data received (9 bytes): [S|A|S|4]
94 This message is not for me.

```

Fichero alarmReg.dat generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1310564064	S	41.2061	1.7300	87
1	1310564064	S	41.2061	1.7300	87
2	1310564070	S	41.2061	1.7300	87
3	1310564077	S	41.2061	1.7300	87
4	1310564082	S	41.2061	1.7300	87

Todas las alarmas han llegado correctamente. Al igual que se comprobó en la simulación (ver Subsección 5.4.3), los nodos de la red son capaces de descubrir rutas hacia otros no vecinos y, de igual manera, hacerles llegar alarmas y acuses de recibo. Por lo tanto, la red soporta el multisalto. De esta prueba, únicamente, se destacan dos situaciones:

- Se ha registrado una pérdida temporal de la comunicación entre los nodos D y A, probablemente debida a la variabilidad del canal radio. Este efecto ha provocado la pérdida de un acuse de recibo. Esta situación ya se contempló en el desarrollo del software y, como se ha visto, el sistema la soporta sin mayor problema. Al no

recibir el nodo S su acuse de recibo tras ALARM\_TIMEOUT, como en la configuración ALARM\_RETRIES es dos, se reintenta el envío de la alarma, en este caso la 1. La segunda vez todo va correctamente. Por este motivo dicha alarma aparece repetida en el fichero `alarmReg.dat` generado por el nodo D.

- En cierto momento, el nodo D ha recibido un mensaje del S, que inicialmente se intentó colocar lo suficientemente separado para que esto no ocurriera. Al ser un mensaje unicast dirigido al nodo A el D lo ha descartado sin más. De este caso se destaca que el área de cobertura varía y, por lo tanto, dos nodos situados a una distancia próxima al límite de éste pueden tener una comunicación intermitente. Para garantizar un comportamiento más estable habrá que dejar cierto margen respecto al límite.

### 6.3. Segunda prueba

Ha consistido en el envío de 4 alarmas desde el nodo S y otras 4 desde el A. Además, durante el proceso de envío de las alarmas, el nodo D ha realizado una solicitud de posición mediante el envío en broadcast de un mensaje PREQ.

Nuevamente se han registrado los mensajes del nodo D para posteriormente analizar lo sucedido durante la prueba.

```

1 // Lee del puerto serie un RREQ del nodo A y otro del S y los pone en
2 // la cola de mensajes.
3 Serial data received (48 bytes): [0|Q|A|1310570952|D|A|1]
4 [0|Q|S|1310578153|D|A|2]
5
6 // Coge de la cola de mensajes el RREQ del nodo A, guarda la ruta y
7 // responde con un RREP unicast.
8 RREQ message received!
9 Unicast RREP message (24 bytes): [A|P|D|D|1310570953|1|A]
10 ****
11 ***          ROUTING TABLE          ***
12 *-----*
13 *   DNID      NNID      SEQN      HOPS      *
14 *-----*
15 *   A          A          1310570952    1          *
16 ****
17 ***          RREQ TABLE          ***
18 *-----*
19 *   SNID      SeqNum      *
20 *-----*
21 *   A          1310570952    *
22 ****
23 // Coge de la cola de mensajes el RREQ del nodo S, guarda la ruta y
24 // responde con un RREP unicast.
25 RREQ message received!
26 Unicast RREP message (24 bytes): [A|P|D|D|1310578154|1|S]
27 ****
28 ***          ROUTING TABLE          ***
29 *-----*
  *   DNID      NNID      SEQN      HOPS      *
```

```

30 *-----*
31 * A A 1310570952 1 *
32 * S A 1310578153 2 *
33 ****
34 ****
35 *** RREQ TABLE ***
36 *-----*
37 * SNID SeqNum *
38 *-----*
39 * A 1310570952 *
40 * S 1310578153 *
41 ****
42
43 // Lee del puerto serie la primera alarma de los nodos A y S y las
   pone en la cola de mensajes.
44 Serial data received (84 bytes):
45 [D|D|D|A|1|1310564064|A|41.2061|1.7300|87]
46 [D|D|D|S|1|1310564064|S|41.2061|1.7300|87]
47
48 // Coge la primera alarma del nodo A de la cola de mensajes, la
   registra y manda su correspondiente ACK.
49 DATA message received!
50 Alarm received: alID: 1 timeStamp: 1310564064 alType: A
51           Lat: 41.2061 Lon: 1.7300 PerConf: 87
52 Unicast ACK message (9 bytes): [A|A|A|1]
53
54 // Coge la primera alarma del nodo S de la cola de mensajes, la
   registra y manda su correspondiente ACK.
55 DATA message received!
56 Alarm received: alID: 1 timeStamp: 1310564064 alType: S
57           Lat: 41.2061 Lon: 1.7300 PerConf: 87
58 Unicast ACK message (9 bytes): [A|A|S|1]
59
60 // El nodo D manda un mensaje de solicitud de posición, PREQ.
61 PREQ message sent (22 bytes): [0|W|D|1310578155|D|1]
62
63 // Lee del puerto serie la segunda alarma del nodo A y el ACK de la
   primera alarma del S reenviado por el nodo A.
64 Serial data received (51 bytes):
65 [D|D|D|A|2|1310564070|A|41.2061|1.7300|87]
66 [S|A|S|1]
67
68 // Coge la segunda alarma del nodo A de la cola de mensajes, la
   registra y manda su correspondiente ACK.
69 DATA message received!
70 Alarm received: alID: 2 timeStamp: 1310564070 alType: A
71           Lat: 41.2061 Lon: 1.7300 PerConf: 87
72 Unicast ACK message (9 bytes): [A|A|A|2]
73
74 // Coge el ACK de la primera alarma del nodo S reenviado por el A y
   lo descarta.
75 This message is not for me.
76
77 // Lee del puerto serie la segunda alarma del nodo S, el PREP del A y
   el reenvío del PREQ hecho también por el nodo A.
78 Serial data received (103 bytes):

```

```
79      [D|D|D|S|2|1310564070|S|41.2061|1.7300|87]
80      [D|C|D|1310570975|A|+90.1234|-127.4321]
81      [O|W|D|1310578155|A|2]
82
83 // Coge la segunda alarma del nodo S de la cola de mensajes, la
   registra y manda su correspondiente ACK.
84 DATA message received!
85 Alarm received: alID: 2           timeStmp: 1310564070      alType: S
86                      Lat: 41.2061      Lon: 1.7300                  PerConf: 87
87 Unicast ACK message (9 bytes): [A|A|S|2]
88
89 // Coge el PREP del nodo A de la cola de mensajes y lo registra.
90 PREP message received!
91 PREP message received: TS: 1310570975      NID: A
92                      LAT: +90.1234      LON: -127.4321
93
94 // Coge el reenvío del PREQ hecho por el nodo A, y no hace nada ya
   que él es el nodo que hizo el PREQ.
95 PREQ message received!
96
97 // Lee del puerto serie la tercera alarma del nodo A y el ACK de la
   segunda alarma del S reenviado por el nodo A.
98 Serial data received (51 bytes):
99      [D|D|D|A|3|1310564077|A|41.2061|1.7300|87]
100     [S|A|S|2]
101
102 // Coge la tercera alarma del nodo A de la cola de mensajes, la
   registra y manda su correspondiente ACK.
103 DATA message received!
104 Alarm received: alID: 3           timeStmp: 1310564077      alType: A
105                      Lat: 41.2061      Lon: 1.7300                  PerConf: 87
106 Unicast ACK message (9 bytes): [A|A|A|3]
107 // Coge el ACK de la segunda alarma del nodo S reenviado por el A y
   lo descarta.
108 This message is not for me.
109
110 // Lee del puerto serie la tercera alarma del nodo S y la cuarta
   alarma del A
111 Serial data received (84 bytes):
112      [D|D|D|S|3|1310564077|S|41.2061|1.7300|87]
113     [D|D|D|A|4|1310564082|A|41.2061|1.7300|87]
114
115 // Coge la tercera alarma del nodo S de la cola de mensajes, la
   registra y manda su correspondiente ACK.
116 DATA message received!
117 Alarm received: alID: 3           timeStmp: 1310564077      alType: S
118                      Lat: 41.2061      Lon: 1.7300                  PerConf: 87
119 Unicast ACK message (9 bytes): [A|A|S|3]
120
121 // Coge la cuarta alarma del nodo A de la cola de mensajes, la
   registra y manda su correspondiente ACK.
122 DATA message received!
123 Alarm received: alID: 4           timeStmp: 1310564082      alType: A
124                      Lat: 41.2061      Lon: 1.7300                  PerConf: 87
125 Unicast ACK message (9 bytes): [A|A|A|4]
```

```

127 // Lee del puerto serie el ACK de la tercera alarma del nodo S
128     reenviado por el A y lo descarta.
129 Serial data received (9 bytes): [S|A|S|3]
130 This message is not for me.
131
132 // Lee del puerto serie la cuarta alarma del nodo S, la registra y
133     manda su correspondiente ACK.
134 Serial data received (42 bytes):
135     [D|D|D|S|4|1310564082|S|41.2061|1.7300|87]
136 DATA message received!
137 Alarm received: alID: 4          timeStamp: 1310564082      alType: S
138             Lat: 41.2061      Lon: 1.7300                  PerConf: 87
139 Unicast ACK message (9 bytes): [A|A|S|4]
140
141 // Lee del puerto serie el ACK de la cuarta alarma del nodo S
142     reenviado por el A y lo descarta.
143 Serial data received (9 bytes): [S|A|S|4]
144 This message is not for me.
145
146 // Vuelve a recibir la cuarta alarma del nodo S, la registra de nuevo
147     y vuelve a mandar su correspondiente ACK. Recibir esta alarma de
148     nuevo significa que el ACK que reenvió el nodo A no llegó al S, y
149     éste tras esperar ALARM_TIMEOUT volvió a mandar la alarma 4.
150 Serial data received (42 bytes):
151     [D|D|D|S|4|1310564082|S|41.2061|1.7300|87]
152 DATA message received!
153 Alarm received: alID: 4          timeStamp: 1310564082      alType: S
154             Lat: 41.2061      Lon: 1.7300                  PerConf: 87
155 Unicast ACK message (9 bytes): [A|A|S|4]
156
157 // Recibe el ACK de la cuarta alarma del nodo S reenviado por el A y
158     lo descarta.
159 Serial data received (9 bytes): [S|A|S|4]
160 This message is not for me.
161
162 // Tras un periodo de tiempo sin recibir nada, puede suceder que el
163     nodo A se haya caído y el S y D no puedan comunicarse, o que el
164     los nodos hayan terminado de enviar sus alarmas y estén a la
165     espera de que aparezcan nuevas.

```

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1310564064	A	41.2061	1.7300	87
1	1310564064	S	41.2061	1.7300	87
2	1310564070	A	41.2061	1.7300	87
2	1310564070	S	41.2061	1.7300	87
3	1310564077	A	41.2061	1.7300	87
3	1310564077	S	41.2061	1.7300	87
4	1310564082	A	41.2061	1.7300	87
4	1310564082	S	41.2061	1.7300	87
4	1310564082	S	41.2061	1.7300	87

Fichero `positionReg.dat` generado por el nodo D:

timeStamp	SNID	latitude	longitude
1310570975	A	+90.1234	-127.4321

En este caso, también han llegado todas las alarmas correctamente al nodo colector, sin embargo, la información de posición del nodo S no lo ha hecho. Se destacan los siguientes sucesos:

- En todos los casos se reciben primero las alarmas del nodo A, que es el más cercano al colector. En caso de existir más nodos, o de haber rutas con más saltos, este efecto puede verse aumentado, pudiendo influir en el correcto funcionamiento de la herramienta. De todos modos, como ya se ha reseñado varias veces en la memoria, el sistema de alarmas de eventos acústicos avisará de la presencia de la fuente sonora en un área y, por lo tanto, el número de alarmas y la frecuencia de aparición de las mismas no se espera que sea tan elevado.
- El nodo D no ha recibido el PREP del S. Mirando con detalle el flujo de mensajes, en la línea 81, vemos que el nodo A reenvía correctamente el PREQ, por consiguiente, el mensaje se ha perdido entre este nodo y el S. Nuevamente se vuelve a detectar una caída temporal de la comunicación. En este caso, para obtener la información de localización del nodo S, el colector debería volver a enviar el PREQ con el gasto de recursos que ello supone.
- La cuarta alarma del nodo S llega repetida debido a la pérdida de un ACK, concretamente entre el S y el A (línea 140).

## 6.4. Tercera prueba

Como se comentó en la Subsección 2.5.2, Yapicioglu y Oktug (2009) proponían en las conclusiones de su estudio sobre la influencia de la altura de las olas en las redes de sensores en el mar la utilización de boyas únicamente enrutadoras o repetidoras, sin sistema sensor. En esta prueba se ha desconectado el ordenador portátil del nodo A y se ha configurado éste en modo *loopback*, tal y como se hizo para el estudio de alcance. De esta manera, el nodo únicamente actúa como repetidor de todos los mensajes que reciba.

En la prueba el nodo S envía 4 alarmas al colector. A continuación, se muestran los mensajes que este último ha generado en el proceso.

```

1 // El nodo D recibe un RREQ del S, pero el contador de saltos se
2     encuentra a 1. Esto se debe a que este mensaje a sido repetido por
3     el nodo A, sin haberle incrementado el contador de saltos. El
4     nodo D guarda la ruta, y responde con un RREP.
5 Serial data received (24 bytes): [0|Q|S|1310578564|D|S|1]
6 RREQ message received!
7 Unicast RREP message (24 bytes): [S|P|D|D|1310578565|1|S]
8 ****
9 ***          ROUTING TABLE          ***
10 *-----*
11 *   DNID      NNID      SEQN      HOPS    *
12 *-----*
13 *   S          S        1310578564    1        *
14 ****
15 ***          RREQ TABLE          ***

```

```

13 *-----*
14 *      SNID          SeqNum      *
15 *-----*
16 *      S          1310578564      *
17 ****
18 // El nodo D recibe su propio RREP cuando éste es repetido por el
   nodo A, pero lo descarta ya que este tipo de mensaje es unicast y
   su DNID es el nodo S.
19 Serial data received (24 bytes): [S|P|D|D|1310578565|1|S]
20 This message is not for me.
21
22 // Recibe la primera alarma del nodo S, la registra y responde con el
   correspondiente ACK.
23 Serial data received (42 bytes):
24   [D|D|D|S|1|1310564064|S|41.2061|1.7300|87]
25 DATA message received!
26 Alarm received: alID: 1           timeStamp: 1310564064      alType: S
27           Lat: 41.2061     Lon: 1.7300                      PerConf: 87
28 Unicast ACK message (9 bytes): [S|A|S|1]
29
30 // Repetición del ACK por el nodo A descartada.
31 Serial data received (9 bytes): [S|A|S|1]
32 This message is not for me.
33
34 // Recibe la segunda alarma del nodo S, la registra y responde con el
   correspondiente ACK.
35 Serial data received (42 bytes):
36   [D|D|D|S|2|1310564070|S|41.2061|1.7300|87]
37 DATA message received!
38 Alarm received: alID: 2           timeStamp: 1310564070      alType: S
39           Lat: 41.2061     Lon: 1.7300                      PerConf: 87
40 Unicast ACK message (9 bytes): [S|A|S|2]
41
42 // En este caso el nodo A no ha repetido el ACK. Esto significa que
   no lo ha recibido por una pérdida temporal de la comunicación.
   Tras ALARM_TIMEOUT el nodo S vuelve a enviar su segunda alarma.
   Ésta es repetida por el nodo A y, finalmente, la recibe el D. La
   registra nuevamente y responde con el correspondiente ACK.
43 Serial data received (42 bytes):
44   [D|D|D|S|2|1310564070|S|41.2061|1.7300|87]
45 DATA message received!
46 Alarm received: alID: 2           timeStamp: 1310564070      alType: S
47           Lat: 41.2061     Lon: 1.7300                      PerConf: 87
48 Unicast ACK message (9 bytes): [S|A|S|2]
49
50 // Repetición del ACK por el nodo A descartada.
51 Serial data received (9 bytes): [S|A|S|2]
52 This message is not for me.
53
54 // Recibe la tercera alarma del nodo S, la registra y responde con el
   correspondiente ACK.
55 Serial data received (42 bytes):
56   [D|D|D|S|3|1310564077|S|41.2061|1.7300|87]
57 DATA message received!
58 Alarm received: alID: 3           timeStamp: 1310564077      alType: S
59           Lat: 41.2061     Lon: 1.7300                      PerConf: 87

```

```

60 Unicast ACK message (9 bytes): [S|A|S|3]
61
62 // Repetición del ACK por el nodo A descartada.
63 Serial data received (9 bytes): [S|A|S|3]
64 This message is not for me.
65 // Recibe la cuarta alarma del nodo S, la registra y responde con el
   correspondiente ACK.
66 Serial data received (42 bytes): [D|D|D|S|4|1310564082|S
   |41.2061|1.7300|87]
67 DATA message received!
68 Alarm received: alID: 4           timeStamp: 1310564082      alType: S
69                      Lat: 41.2061      Lon: 1.7300                  PerConf: 87
70 Unicast ACK message (9 bytes): [S|A|S|4]
71
72 // Repetición del ACK por el nodo A descartada.
73 Serial data received (9 bytes): [S|A|S|4]
74 This message is not for me.

```

Fichero `alarmReg.dat` generado por el nodo D:

alarm ID	timeStamp	alarmType	latitude	longitude	perConf
1	1310564064	S	41.2061	1.7300	87
2	1310564070	S	41.2061	1.7300	87
2	1310564070	S	41.2061	1.7300	87
3	1310564077	S	41.2061	1.7300	87
4	1310564082	S	41.2061	1.7300	87

El nodo D ha recibido todas las alarmas de S correctamente. Se ha vuelto a detectar una pérdida temporal de la comunicación, teniendo que reenviarse una alarma tras el `ALARM_TIMEOUT` correspondiente.

Utilizar boyas repetidoras puede ser una buena solución para aumentar la distancia entre boyas sensoras, en el caso de que eso interese. Estas boyas requerirán muy poca electrónica; se eliminará el subsistema sensor y el módulo de procesado de audio del computacional, así podrá utilizarse un subsistema de energía reducido.

## 6.5. Discusión de resultados

El objetivo de la realización de esta prueba final ha sido verificar el correcto funcionamiento de todas las partes de este proyecto integradas: módulo de comunicación del subsistema radio y módulo de entrega remota de datos y protocolo de encaminamiento AODV-LAB del subsistema computacional. El resultado, como se ha ido comentado en cada una de las pruebas, ha sido positivo. Únicamente se destaca la pérdida de algún paquete, suceso común en los sistemas de comunicación inalámbrica. De este modo, queda implementada una herramienta o prototipo a partir del cual se puede continuar experimentando y trabajando para ir logrando cada vez una solución mejor.

En base a los resultados obtenidos se pueden extraer una serie de conclusiones más concretas:

- El proceso de Descubrimiento de Ruta con 3 nodos funciona correctamente.
- La red soporta multisalto y realiza adecuadamente la entrega de alarmas.

- Las pérdidas temporales de comunicación entre nodos debido a la variabilidad del canal radio están presentes en todos los casos. En la mayoría de las pruebas realizadas la capacidad de reenvío de alarmas, configurable a nivel de software, permite solventar estos problemas sin un consumo elevado de recursos. Existen algunas excepciones, como es el caso de la pérdida de mensajes de posición, en las que, si se produce un fallo, la nueva solicitud debe hacerse manualmente por el usuario. En este caso, reenviar un PREQ implica que nuevamente todos los nodos que lo reciban envíen su PREP, siendo redundante para todos aquellos que la primera vez lo hicieron correctamente.
- El área de cobertura de un nodo es variable. Por ello, a la hora de colocar los nodos habrá que dejar un determinado margen respecto al límite medido.
- La operación con nodos repetidores en la red es viable y permite aumentar la distancia entre boyas de forma económica.

Para pruebas futuras se considera apropiado:

- Utilizar una población de nodos algo mayor. Permitirá realizar pruebas de descubrimiento de rutas óptimas, y analizar en profundidad el efecto del número de saltos (**MAX\_NUM\_HOPS**) entre un nodo fuente y el colector.
- Estudiar la opción del firmware de utilizar repeticiones a nivel de MAC. Esta característica puede disminuir el efecto de la pérdida temporal de comunicación entre nodos, ya sea: reduciendo la carga computacional que implica, el número de paquetes redundantes que se introducen a la red, o el tiempo empleado en solventar la pérdida de mensajes. Concretamente, las repeticiones a nivel de MAC parecen más apropiadas para pérdidas breves de la comunicación y los reintentos a nivel de red para períodos más largos sin conexión. Debe prestarse atención a la restricción del ciclo de trabajo que existe en la banda, ya que estas repeticiones estarán introduciendo más tráfico al medio.
- Realizar pruebas en el mar. Podrán detectarse efectos nuevos a los registrados en tierra, o de mayor magnitud. También permitirá cuantificar la variabilidad del área de cobertura de los nodos.
- Utilizar un tráfico de datos real o acorde con el que generará el sistema de alarmas.
- Estudiar en detalle las opciones de configuración de los nodos y ver su efecto en la red para lograr un funcionamiento más óptimo.

# Capítulo 7

## Conclusiones y trabajo futuro

*Parecía que habíamos llegado al final del camino y resulta que era sólo una curva abierta a otro paisaje y a nuevas curiosidades.*

El año de la muerte de Ricardo Reis  
José Saramago

### 7.1. Conclusiones

Este proyecto ha tenido como meta el dar una respuesta a la necesidad manifiesta de aportar soluciones viables y económicamente sostenibles para la utilización de redes inalámbricas de sensores en el medio marino, orientadas a la monitorización de cetáceos, fuentes acústicas, u otras aplicaciones.

Se ha tratado de abarcar los diferentes campos implicados en este ámbito de actuación con el objetivo de formar una base sólida y completa sobre la que continuar trabajando.

Atendiendo a los objetivos definidos en el primer capítulo de la presente memoria se extraen las siguientes conclusiones:

1. La implementación de redes ad hoc con capacidad sensora en el medio marino es viable y sostenible económicamente utilizando tecnologías estandarizadas, o con elevada presencia en el mercado.

En función de la aplicación, o servicio que deba soportar la red, se optará por una u otra tecnología teniendo en cuenta que:

- Existe un compromiso entre velocidad de envío de datos y alcance de la señal.
- Cuanto menor sea el consumo energético del sistema de comunicación menor será el coste final de cada nodo.

En este proyecto se optó por el desarrollo de una red capaz de dar soporte a un servicio de alarmas, lo que supone un flujo bajo de datos y permite maximizar la distancia entre nodos.

2. Se ha decidido operar en la banda ICM de 869.40 a 869.65 MHz, donde: está permitido emplear una PIRE considerablemente superior a otras bandas, existen módulos económicos y de bajo consumo, y el compromiso entre velocidad de envío

de datos y alcance permite la implementación de una red ad hoc y distancias entre nodos de algunos kilómetros. Concretamente se ha optado por los módulos XBee-PRO 868.

3. La solución propuesta responde a una arquitectura modular, lo que ha facilitado su integración con el resto del sistema LIDO DCL ya desarrollado.
4. Se ha realizado un estudio teórico de un enlace radioeléctrico en entorno marino empleando el modelo de tierra plana. El resultado obtenido ha sido que el alcance teórico para los módulos XBee-PRO 868 es de 5.6 kilómetros. Finalmente, se realizaron medidas reales sobre la superficie del mar logrando buenas prestaciones a distancias de hasta aproximadamente 2 kilómetros, suficiente para el servicio de alarmas.

La desviación entre el alcance teórico y el empírico se debe a que el modelo de tierra plana supone el mar como una superficie lisa y no contempla otros factores que están presentes en el caso real, como: pérdidas de potencia en los equipos, variación de la altura relativa de las antenas, difracción por barcos u el propio oleaje, atenuación por gases o partículas en suspensión, etc.

5. Se ha desarrollado en C/Linux un módulo de entrega remota de datos, encargado de la gestión de las alarmas en el nodo local y de su correcto envío al colector. Este módulo emplea el protocolo de encaminamiento AODV-LAB, también implementado en el proyecto.
6. Se ha simulado la red recreando varios escenarios y situaciones distintas para verificar el correcto funcionamiento del software desarrollado. Los resultados obtenidos confirmaron las prestaciones deseadas.

Todo esto integrado ha dado lugar a un prototipo de red ad hoc de datos que servirá como herramienta para continuar experimentando y trabajando en esta línea. Por ello, se considera cumplido el objetivo general planteado para este proyecto.

## 7.2. Trabajo futuro

A lo largo de esta memoria se han apuntado las siguientes líneas de actuación orientadas a la mejora y continuación del proyecto:

1. **Realización de más medidas de alcance sobre la superficie el mar.** Emplear los módulos XBee-PRO 868 situados en boyas o, al menos, en una embarcación. Tratar de aproximarse lo máximo posible a la situación final del sistema. Tomar datos durante un trayecto pudiendo registrar más claramente el efecto del aumento de la distancia en el enlace. Repetir las medidas en tierra firme para determinar la magnitud de efectos como: la reflexión en el agua salada, la variación de la altura relativa de las antenas, la atenuación por partículas en suspensión, etc.
2. **Evaluación de otras tecnologías de comunicación.** Realizar medidas en entorno marino con equipos que operen en otras bandas de frecuencia. Elaborar un informe de medidas y resultados empíricos con diferentes tecnologías que permita plantearse el desarrollo de nuevas aplicaciones, o servicios para la mitigación del

impacto de la contaminación acústica en el medio marino y otras herramientas para el estudio de este medio. Se propone la utilización de: módulos RF en la banda 433 MHz, Wi-Fi en 2.4 y 5.6 GHz, y WiMAX en 5.6 GHz.

3. **Realización de un seguimiento del mercado.** Conviene estar atento a la evolución de los módulos de comunicación. Las redes de sensores y ad hoc son un tema muy activo de investigación en varias universidades, y cada vez hay más aplicaciones comerciales basadas en ellas. Constantemente aparecen en el mercado módulos con nuevas prestaciones. Concretamente en la banda 868 MHz están empezando a aparecer unidades con soporte de topología mesh, lo que simplifica considerablemente el desarrollo de aplicaciones.
4. **Profundización en el estudio teórico de enlaces radioeléctricos en el mar.** Ahondando en este estudio se podrá lograr un modelo teórico más fiel a la realidad que el utilizado en el proyecto. Deberá tener en cuenta, entre otros, los efectos comentados en el punto 4 de las conclusiones, cuantificando su magnitud.
5. **Evaluación de otros protocolos de encaminamiento.** Esta evaluación permitirá realizar una elección más precisa según la aplicación que interese. Para ello se pueden emplear simuladores de redes, como NS-2, obteniendo resultados cuantitativos de varios protocolos de encaminamiento que ayudarán a establecer una comparativa.
6. **Módulo de entrega remota de datos.** En este ámbito se podrían añadir nuevas funcionalidades, mejoras, correcciones y trabajar en la compresión de datos para reducir el tráfico de la red, entre otros.
7. **Realización de pruebas reales de operación de la red.** Estas pruebas permitirán:
  - Evaluar distintas configuraciones con un número más elevado de nodos.
  - Estudiar el efecto en la red de los diferentes parámetros de configuración de los módulos.
  - Realizar pruebas en el mar con un tráfico real de alarmas.

Se considera importante continuar con estas líneas de investigación con el objetivo de mitigar la problemática medioambiental comentada en el Capítulo 1, a través de soluciones tecnológicas viables y sostenibles en todos sus aspectos.



## Parte II

# Apéndices



# Apéndice A

## Presupuesto

### EJECUCIÓN MATERIAL

Compra de ordenador personal (software incluido)	2.000 €
Alquiler impresora láser durante 6 meses	260 €
Material oficina	300 €
Material técnico	
Módulo XBee-PRO 868 (x3) (Venco Electrónica)	180 €
Conversor UART-USB (x3) (Lextronic)	70 €
Antena 4.5 dBi (x3) (Cooking-Hacks)	30 €
Cable miniUSB-USB (x3) (Cooking-Hacks)	10 €
TOTAL EJECUCIÓN MATERIAL	2.850 €

### GASTOS GENERALES

13 % sobre Ejecución Material	370,5 €
-------------------------------	---------

### BENEFICIO INDUSTRIAL

6 % sobre Ejecución Material	171 €
------------------------------	-------

### HONORARIOS PROYECTO

1500 horas a 15 €/ hora	22.500 €
-------------------------	----------

### MATERIAL FUNGIBLE

Gastos de impresión	150 €
Encuadernación	10 €
TOTAL MATERIAL FUNGIBLE	160 €

### PRESUPUESTO TOTAL

Presupuesto total sin I.V.A.	26.051,5 €
Presupuesto total con I.V.A. (18 %)	30.740,8 €

Barcelona, Septiembre de 2011  
El Ingeniero Jefe de Proyecto

Fdo.: Ramiro Utrilla Gutiérrez  
Ingeniero Superior de Telecomunicación



## Apéndice B

# Código MATLAB

En este apéndice se incluye el código MATLAB utilizado en el Capítulo 4 para el estudio teórico de un enlace radioeléctrico en el mar.

### B.1. freeSpace.m

```
1 %-----%
2 % Ramiro Utrilla Gutiérrez %
3 % Julio 2011 %
4 %-----
5 close all
6 clear all
7 clc
8
9 % RADIO LINK - FREE SPACE PROPAGATION
10 % Definitions
11 d = [1000:100:1000*10^3]; % m
12 freq = 869.525 * 10^6; % Hz
13 speed = 3 * 10^8; % m/s
14 lambda = speed / freq; % m
15 Pt = 25; % dBm
16 Gt = 4.5; % dBi
17 Gr = 4.5; % dBi
18 Pr = zeros(1,length(d)); % dBm
19 sens = -104*ones(1,length(d)); % dBm
20
21 % Free Space Path Loss
22 Lbf = zeros(1,length(d));
23 Lbf = 32.45 + 20*log10(d/(10^3)) + 20*log10(freq/(10^6)); % dB
24
25 % Receive Power
26 Pr = Pt + Gt - Lbf + Gr; % dBm
27
28 % Figure 1. Free Space Path Loss
29 figure
30 semilogx(d/(10^3), Lbf, 'b', 'linewidth', 2);
31 grid on
32 set(gca, 'XTickLabel', {'1', '10', '100', '1.000'})
33 xlabel('\bf{Distancia [km]}')
34 ylabel('\bf{L_{bf}} [dB]')
```

```

35 title(sprintf('Pérdida básica de propagación en condiciones de
36 espacio libre \n Frecuencia de operación: 869.525 MHz'));
37
38 % Figure 2. Receive Power
39 figure
40 semilogx(d/(10^3), Pr, 'b', 'linewidth', 2);
41 grid on
42 set(gca,'XTickLabel',[1,10,100,1.000])
43 xlabel('\bfDistancia [km]')
44 ylabel('\bfP_{r} [dBm]')
45 title(sprintf('Potencia disponible en el receptor en condiciones de
46 espacio libre \n P_{t}: %d dBm; G_{t}: %g dBi; G_{r}: %g dBi', Pt,
47 Gt, Gr));
48 hold on
49 semilogx(d/(10^3), sens, 'r', 'linewidth', 2);
50 legend('Potencia disponible en el receptor','Sensibilidad del
51 receptor')
52 hold off

```

## B.2. twoRayModel.m

```

1 %-----%
2 % Ramiro Utrilla Gutiérrez %
3 % Julio 2011 %
4 %-----%
5 clear all
6 close all
7 clc
8
9 % RADIO LINK - TWO RAY PROPAGATION MODEL
10 % Definitions
11 d = [1:.01:10000]; % m
12 f = 869.525 * 10^6; % Hz
13 c = 3 * 10^8; % m/s
14 lambda = c/f; % m
15 Pt = 25; % dBm
16 Gt = 4.5; % dBi
17 Gr = 4.5; % dBi
18 ht = 2; % m
19 hr = 2; % m
20 beta = pi; % rad
21 modR = 1;
22 lb = zeros(1,length(d));
23 Lb = zeros(1,length(d)); % dB
24 Lbf = zeros(1,length(d)); % dB
25 Lb2 = zeros(1,length(d)); % dB
26 Pr = zeros(1,length(d)); % dBm
27 sens = -104*ones(1,length(d)); % dBm
28
29 % Two Ray Model Path Loss
30 delta = (4*pi*ht*hr)./(lambda*d);
31 for i=1:length(d)
32     lb(i) = ((4*pi*d(i))/lambda)^2 / (2*(1-cos(delta(i))));
33     Lb(i) = 10*log10(lb(i));

```

```

34 end
35
36 % Free Space Path Loss
37 Lbf = 32.45 + 20*log10(d/(10^3)) + 20*log10(f/(10^6)); % dBm
38
39 % Two Ray Model Path Loss for long distances
40 Lb2 = 40*log10(d/(10^3)) - 20*log10(ht*hr) + 120;
41
42 % Receive Power
43 Pr = Pt + Gt - Lb + Gr; % dBm
44
45 % Figure 1. Two Ray Path Loss 1
46 figure
47 semilogx(d, Lb, 'linewidth', 2)
48 grid on
49 set(gca,'XTickLabel',{'1','10','100','1.000','10.000'})
50 xlabel('\bf{Distancia [m]}')
51 ylabel('\bf{Pérdida básica de propagación [dB]}')
52 title(sprintf('Pérdida básica de propagación \n f: 869.525 MHz; h_{t
      }: %d m; h_{r}: %d m', ht, hr));
53 hold on
54 semilogx(d, Lbf, 'r', 'linewidth', 2);
55 legend('Modelo de tierra plana','En condiciones de espacio libre',2)
56 hold off
57
58 % Figure 2. Two Ray Path Loss 2
59 figure
60 semilogx(d, Lb, 'linewidth', 2)
61 grid on
62 set(gca,'XTickLabel',{'1','10','100','1.000','10.000'})
63 xlabel('\bf{Distancia [m]}')
64 ylabel('\bf{Pérdida básica de propagación [dB]}')
65 title(sprintf('Pérdida básica de propagación \n f: 869.525 MHz; h_{t
      }: %d m; h_{r}: %d m', ht, hr));
66 hold on
67 semilogx(d, Lb2, 'k', 'linewidth', 2);
68 semilogx(d, Lbf, 'r', 'linewidth', 2);
69 legend('Modelo de tierra plana','Modelo de tierra plana largas
      distancias','En condiciones de espacio libre',2)
70 hold off
71
72 % Figure 3. Receive Power
73 figure
74 semilogx(d/(10^3), Pr, 'b', 'linewidth', 2);
75 grid on
76 set(gca,'XTickLabel',{'1','10','100','1.000','10.000'})
77 xlabel('\bf{Distancia [m]}')
78 ylabel('\bf{P_{r} [dBm]}')
79 title(sprintf('Potencia disponible en el receptor con modelo de
      tierra plana \n f: 869.525 MHz; P_{t}: %d dBm; G_{t}: %g dBi; G_{r
      }: %g dBi; h_{t}: %d m; h_{r}: %d m', Pt, Gt, Gr, ht, hr));
80 hold on
81 semilogx(d/(10^3), sens, 'r', 'linewidth', 2);
82 legend('Potencia disponible en el receptor','Sensibilidad del
      receptor')
83 hold off

```

### B.3. measurements.m

```

1 %----- %
2 % Ramiro Utrilla Gutiérrez %
3 % Julio 2011 %
4 %----- %
5 clear all
6 close all
7 clc
8
9 % RADIO LINK - TWO RAY PROPAGATION MODEL
10 % Definitions
11 d = [1:.001:10]; % m
12 f = 869.525 * 10^6; % Hz
13 c = 3 * 10^8; % m/s
14 lambda = c/f; % m
15 Pt = 25; % dBm
16 Gt = 4.5; % dBi
17 Gr = 4.5; % dBi
18 ht = 2; % m
19 hr = 2; % m
20 beta = pi; % rad
21 modR = 1;
22 lb = zeros(1,length(d));
23 Lb = zeros(1,length(d)); % dB
24 day1e1 = zeros(1,5); % dBm
25 day2e1 = zeros(1,5); % dBm
26 day1e2 = zeros(1,5); % dBm
27 day2e2 = zeros(1,5); % dBm
28 day1e3 = zeros(1,5); % dBm
29 day2e3 = zeros(1,5); % dBm
30 Pr = zeros(1,length(d)); % dBm
31 sens = -104*ones(1,length(d)); % dBm
32 link1 = 1.84*ones(1,5); % km
33 link2 = 3.06*ones(1,5); % km
34 link3 = 3.5*ones(1,5); % km
35
36 % Two Ray Model Path Loss
37 delta = (4*pi*ht*hr)./(lambda*(d*10^3));
38 for i=1:length(d)
39     lb(i) = ((4*pi*(d(i)*10^3))/lambda)^2 / (2*(1-cos(delta(i))));
40     Lb(i) = 10*log10(lb(i));
41 end
42
43 % Receive Power
44 Pr = Pt + Gt - Lb + Gr; % dBm
45
46 % Measurements
47 day1e1 = [-95; -93; -96; -103; -97];
48 day2e1 = [-92; -98; -95; -102; -97];
49
50 % Hypothetical Link 2
51 space1 = Pr(841) - Pr(2061);
52 day1e2 = day1e1 - space1;
53 day2e2 = day2e1 - space1;

```

```

54
55 % Hypothetical Link 2
56 space2 = Pr(841) - Pr(2501);
57 day1e3 = day1e1 - space2;
58 day2e3 = day2e1 - space2;
59
60 % Figure 1. Receive Power and Link 1 Measurements
61 figure
62 semilogx(d, Pr, 'b', 'linewidth', 2);
63 grid on
64 set(gca,'XTickLabel',[1,10])
65 xlabel('Distancia [km]')
66 ylabel('P_{r} [dBm]')
67 title(sprintf('Potencia disponible en el receptor. ENLACE 1 \n f:
869.525 MHz; P_{t}: %d dBm; G_{t}: %g dBi; G_{r}: %g dBi; h_{t}: %
d m; h_{r}: %d m', Pt, Gt, Gr, ht, hr));
68 hold on
69 semilogx(link1, day1e1, 'kx', 'linewidth', 2);
70 semilogx(link1, day2e1, 'mo', 'linewidth', 2);
71 semilogx(d, sens, 'r', 'linewidth', 2);
72 legend('Modelo de tierra plana','Medidas de la jornada 1','Medidas de
la jornada 2','Sensibilidad del receptor')
73 hold off
74
75 % Figure 2. Receive Power and Hypothetical Link 2 Measurements
76 figure
77 semilogx(d, Pr, 'b', 'linewidth', 2);
78 grid on
79 set(gca,'XTickLabel',[1,10])
80 xlabel('Distancia [km]')
81 ylabel('P_{r} [dBm]')
82 title(sprintf('Potencia disponible en el receptor. ENLACE 2 \n f:
869.525 MHz; P_{t}: %d dBm; G_{t}: %g dBi; G_{r}: %g dBi; h_{t}: %
d m; h_{r}: %d m', Pt, Gt, Gr, ht, hr));
83 hold on
84 semilogx(link2, day1e2, 'kx', 'linewidth', 2);
85 semilogx(link2, day2e2, 'mo', 'linewidth', 2);
86 semilogx(d, sens, 'r', 'linewidth', 2);
87 legend('Modelo de tierra plana','Medidas hipotéticas de la jornada 1',
'Medidas hipotéticas de la jornada 2','Sensibilidad del receptor',
)
88 hold off
89
90 % Figure 3. Receive Power and Hypothetical Link 3 Measurements
91 figure
92 semilogx(d, Pr, 'b', 'linewidth', 2);
93 grid on
94 set(gca,'XTickLabel',[1,10])
95 xlabel('Distancia [km]')
96 ylabel('P_{r} [dBm]')
97 title(sprintf('Potencia disponible en el receptor. ENLACE 3 \n f:
869.525 MHz; P_{t}: %d dBm; G_{t}: %g dBi; G_{r}: %g dBi; h_{t}: %
d m; h_{r}: %d m', Pt, Gt, Gr, ht, hr));
98 hold on
99 semilogx(link3, day1e3, 'kx', 'linewidth', 2);
100 semilogx(link3, day2e3, 'mo', 'linewidth', 2);

```

```
101 semilogx(d, sens, 'r', 'linewidth', 2);
102 legend('Modelo de tierra plana','Medidas hipotéticas de la jornada 1',
103 , 'Medidas hipotéticas de la jornada 2','Sensibilidad del receptor')
      )
hold off
```

## Apéndice C

# Cabecera y definición de funciones

En este apéndice se incluyen las cabeceras y declaraciones de las principales funciones implementadas en el sistema.

### C.1. extractMessInfo

```
1  ****
2  /* Take out the NNID and frameID headers of the message received */
3  ****
4  /* INPUT
5   * -message:    Message received
6   */
7  /* OUTPUT
8   * -message:    Message received without NNID and frameID fields.
9   * -nnid:        NNID field
10  * -frameID:    frameID field
11  */
12  /* RETURNS
13  * - ERROR if there is any problem in the process.
14  * - OK if the process is succesful.
15  */
16  ****
17  int extractMessInfo (char *nnid, char *frameID, char *message);
```

### C.2. checkAlarm

```
1  ****
2  /* Check if there are new alarms in the alarm.dat file. */
3  ****
4  /* INPUT
5   * -lastAlarm: The alarmID of the last alarm used.
6   */
7  /* RETURNS
8   * -ERROR if there is not more alarms now.
9   * -The index of the line in the file with the next alarm to send*/
10  ****
11  int checkAlarm (unsigned int lastAlarm);
```

### C.3. sendLocalAlarm

```

1  *****/
2  /* Read an alarm in the 'alarmIndex' line from the 'alarm.dat'      */
3  /* file and send it.                                                 */
4  *****/
5  /* INPUT */
6  /* -dnid:          DNID field                                     */
7  /* -nnid:          NNID field                                     */
8  /* -nid:           SNID field                                     */
9  /* -alarmIndex:   The alarm index of the alarm to read.           */
10 /* */
11 /* RETURNS */
12 /* -The alarmID of the alarm sended.                                */
13 /* -0 if there is any problem in the process.                      */
14 *****/
15 unsigned int sendLocalAlarm (char *dnid, char *nnid, char *nid, int
    alarmIndex);

```

### C.4. exeDATA

```

1  *****/
2  /* Execute the DATA reception process.                            */
3  *****/
4  /* INPUT */
5  /* -nid:           NID of the own node.                           */
6  /* -DATAmessage:  DATA message received.                         */
7  /* -lt_rte:        Lifetime of a routing table entry.          */
8  /* */
9  /* OUTPUT */
10 /* -auxdnid:       DNID of the ACK, when a route has to be     */
11 /*                  rediscovered.                                 */
12 /* */
13 /* RETURNS */
14 /* -ERROR if there is any problem in the process.               */
15 /* -OK if the process is succesful.                             */
16 *****/
17 int exeDATA(char *nid, char *DATAmessage, int lt_rte, char *auxdnid);

```

### C.5. sendDATA

```

1  *****/
2  /* Send a unicast DATA message.                                */
3  *****/
4  /* INPUT */
5  /* -nnid:           Next node ID in the route.                 */
6  /* -DATAmessage:   The part of the message with the info.      */
7  *****/
8  void sendDATA (char *nnid, char *DATAmessage);

```

### C.6. exeACK

```
1  ****  
2  /* Execute the ACK reception process. */  
3  ****  
4  /* INPUT  
5   * -nid           NID of the own node. */  
6  /* -ACKmessage:    ACK message received. */  
7  /* -lt_rte:        Lifetime of a routing table entry. */  
8  /* RETURNS  
9   * -alarmID when the ACK messages corresponds with a local */  
10 /* alarm sended. */  
11 /* -0 in other cases. */  
12 ****  
13 unsigned int exeACK (char *nid, char *ACKmessage, int lt_rte);
```

### C.7. sendACK

```
1  ****  
2  /* Send a unicast ACK message. */  
3  ****  
4  /* INPUT */  
5  /* -nnid Next node ID in the route. */  
6  /* -ACKmessage: The part of the message with the info. */  
7  ****  
8  void sendACK (char *nnid, char *ACKmessage);
```

### C.8. serialData2queue

```
1  ****  
2  /* Read the serial data and put the messages received between */  
3  /* brackets ([Hello world!]) in the message queue. */  
4  ****  
5  /* INPUT */  
6  /* -serialData: Data readed in the serial port. */  
7  /* -Q: Message queue. */  
8  ****  
9  void serialData2queue (char *serialData, queue Q);
```

### C.9. initSeqNum

```
1  ****  
2  /* Initialize the sequence number with time(NULL) for making a      */  
3  /* strong system against resets.                                     */  
4  ****  
5  void initSeqNum (void);
```

## C.10. updateSeqNum

```

1  /*************************************************************************/
2  /* Update the sequence number.                                         */
3  /*************************************************************************/
4  /* INPUT                                                               */
5  /* -seqN: The sequence number received in a RREQ or PREQ messages */
6  /* */                                                               */
7  /* RETURNS                                                             */
8  /* -The sequence number updated.                                       */
9  /* */                                                               */
10 unsigned long int updateSeqNum (unsigned long int seqN);

```

## C.11. checkRREQ

```

1  /*************************************************************************/
2  /* Check the RREQ table to verify that the RREQ is updated.          */
3  /*************************************************************************/
4  /* INPUT                                                               */
5  /* -snid: Source node id of the RREQ message.                      */
6  /* -seqN: The sequence number received in a RREQ message.           */
7  /* */                                                               */
8  /* RETURNS                                                             */
9  /* -OK to add the new entry.                                         */
10 /* -ERROR to discard the new entry.                                    */
11 /* */                                                               */
12 int checkRREQ (char *snid, unsigned long int seqN);

```

## C.12. exeRREQ

```

1  /*************************************************************************/
2  /* Execute the RREQ process.                                         */
3  /*************************************************************************/
4  /* INPUT                                                               */
5  /* -nid:                 NID of the own node.                         */
6  /* -max_hops:            Maximum number of hops setted in the node.conf */
7  /* -RREQmessage:         RREQ message received.                      */
8  /* -lt_rte:              Lifetime of a routing table entry.          */
9  /* */                                                               */
10 /* RETURNS                                                            */
11 /* -OK if the process is succesful.                                   */
12 /* -ERROR if there is any problem in the process.                   */
13 /* */                                                               */
14 /* */                                                               */
15 int exeRREQ (char *nid, int max_hops, char *RREQmessage, int lt_rte);

```

### C.13. sendRREQ

```
1  ****  
2  /* Send a broadcast RREQ message. If the sequence number in the */  
3  /* argument is '0', the RREQ is local and the function increase */  
4  /* the local seqNum. If the argument is different of 0, the RREQ */  
5  /* is not local, it is a rebroadcast and it have its own sequence */  
6  /* number. */  
7  ****  
8  /* INPUT */  
9  /* -snid: SNID field. */  
10 /* -seqN: The sequence number of the RREQ, or 0 if it is a local */  
11 /* RREQ which has to be sent. */  
12 /* -dnid: DNID field. */  
13 /* -pnid: PNID field. */  
14 /* -hops: The number of hops of the frame. */  
15 /* */  
16 /* RETURNS */  
17 /* -OK when make a RREQ rebroadcast. */  
18 /* -Sequence number when sends a local RREQ. */  
19 /* ****  
20 unsigned long int sendRREQ (char *snid, unsigned long int seqN, char  
   *dnid, char *pnid, int hops);
```

#### C.14. checkPREQ

```
1  ****  
2  /* Check the PREQ table to verify that the PREQ is updated. */  
3  ****  
4  /* INPUT  
5   * -snid: Source node id of the PREQ message.  
6   * -seqN: The sequence number received in a PREQ message.  
7  */  
8  /* RETURNS  
9   * -OK to add the new entry.  
10  * -ERROR to discard the new entry.  
11  ****  
12 int checkPREQ (char *snid, unsigned long int seqN);
```

### C.15. exePREQ

```
1  ****  
2  /* Execute the PREQ process. */  
3  ****  
4  /* INPUT */  
5  /* -nid: NID of the own node. */  
6  /* -PREQmessage: PREQ message received. */  
7  /* -lt_rte: Lifetime of a routing table entry. */  
8  */
```

```

9  /* RETURNS
10 /* -OK if the process is succesful. */
11 /* -ERROR if there is any problem in the process. */
12 /*****
13 int exePREQ (char *nid, char *PREQmessage, int lt_rte);

```

## C.16. sendPREQ

```

1  *****/
2  /* Send a broadcast PREQ message. If the sequence number in the */
3  /* argument is '0', the PREQ is local and the function increase */
4  /* the local seqNum. If the argument is different of 0, the PREQ */
5  /* is not local, it is a rebroadcast and it have its own sequence */
6  /* number. */
7  *****/
8  /* INPUT */
9  /* -snid: SNID field. */
10 /* -seqN: The sequence number of the PREQ, or 0 if it is a local */
11 /* PREQ which has to be sent. */
12 /* -pnid: PNID field. */
13 /* -hops: The number of hops of the frame. */
14 /*
15 /* RETURNS
16 /* -OK when make a PREQ rebroadcast. */
17 /* -Sequence number when sends a local PREQ. */
18 /*****
19 unsigned long int sendPREQ (char *snid, unsigned long int seqN, char
  *pnid, int hops);

```

## C.17. exeRREP

```

1  *****/
2  /* Execute the RREP process. */
3  *****/
4  /* INPUT */
5  /* -nid: NID of the own node. */
6  /* -max_hops: Maximum number of hops setted in the node.conf */
7  /* -RREPmessage: RREP message received. */
8  /* -lt_rte: Lifetime of a routing table entry. */
9  /*
10 /* RETURNS
11 /* -The sequence number (seqN > 0) when the RREP corresponds
12 /* with one local RREQ. */
13 /* -0 in the other cases. */
14 /*****
15 unsigned long int exeRREP (char *nid, int max_hops, char *RREPmessage
  , int lt_rte);

```

## C.18. sendRREP

```

1  /*************************************************************************/
2  /* Send a unicast RREP message. */
3  /*************************************************************************/
4  /* INPUT */
5  /* -nnid: NNID field. */
6  /* -pnid: PNID field. */
7  /* -snid: SNID field. */
8  /* -seqN: The sequence number. */
9  /* -hops: The number of hops of the frame. */
10 /* -dnid: DNID field. */
11 /*************************************************************************/
12 void sendRREP (char *nnid, char *pnid, char *snid, unsigned long int
    seqN, int hops, char *dnid);

```

## C.19. exePREP

```

1  /*************************************************************************/
2  /* Execute the PREP process. */
3  /*************************************************************************/
4  /* INPUT */
5  /* -nid: NID of the own node. */
6  /* -PREPmessage: PREP message received. */
7  /* -lt_rte: Lifetime of a routing table entry. */
8  /* RETURNS */
9  /* -OK if the process is successful. */
10 /* -ERROR if there is any problem in the process. */
11 /*************************************************************************/
12 int exePREP (char *nid, char *PREPmessage, int lt_rte);

```

## C.20. sendPREP

```

1  /*************************************************************************/
2  /* Send a unicast PREP message. */
3  /*************************************************************************/
4  /* INPUT */
5  /* -nnid: NNID field. */
6  /* -dnid: DNID field. */
7  /* -timeStamp: Timestamp in the PREP received, or 0 if it is a */
8  /* local PREP message. */
9  /* -snid: SNID field. */
10 /* -latitude: Latitude in the PREP received, or NULL if it is a */
11 /* local PREP message. */
12 /* -longitude: Longitude in the PREP received, or NULL if it is a */
13 /* local PREP message. */
14 /* */
15 void sendPREP (char *nnid, char *dnid, unsigned long int timeStamp,
    char *snid, char *latitude, char *longitude);

```

## C.21. checkNewRoute

```

1  ****
2  /* Check if the new route is already in the routing table and */
3  /* also check the update conditions. */
4  ****
5  /* INPUT */
6  /* -dnid: Destination node of the route. */
7  /* -seqn: Sequence number of the route. */
8  /* -hops: The number of hops of the route till the own node. */
9  /* -lt_rte: Lifetime of a routing table entry. */
10 /* RETURNS */
11 /* - OK when the routing table is OK to be updated. */
12 /* - ERROR if the routing table is already updated. */
13 ****
14 int checkNewRoute (char *dnid, unsigned long int seqn, int hops, int
15   lt_rte);

```

## C.22. newRoute

```

1  ****
2  /* Add a new route to the routing table. */
3  ****
4  /* INPUT */
5  /* -dnid: Destination node of the route. */
6  /* -nnid: Next node in the route to the DNID. */
7  /* -seqn: Sequence number of the route. */
8  /* -hops: The number of hops of the route till the own node. */
9  /* RETURNS */
10 /* -OK if the process is succesful. */
11 /* -ERROR if there is any problem in the process. */
12 ****
13 int newRoute(char *dnid,char *nnid,unsigned long int seqn,int hops);

```

## C.23. getRoute

```

1  ****
2  /* Look for a valid route to the dnid. */
3  ****
4  /* INPUT */
5  /* -dnid: Destination node of the route. */
6  /* -lt_rte: Lifetime of a routing table entry. */
7  /* OUTPUT */
8  /* -nnid: Next node in the route to the DNID. */
9  /* RETURNS */
10 /* -OK if the process is succesful. */
11 /* -ERROR if there is any problem in the process. */
12 ****
13 int getRoute (char *dnid, int lt_rte, char **nnid);

```

# Bibliografía

*Y así, del mucho leer y del poco dormir, se le secó  
el celebro de manera que vino a perder el juicio.*

Miguel de Cervantes Saavedra

ALBENTIA-SYSTEMS. *Radioenlaces en entornos marinos. Application note*. Albentia-Systems, 2008. Disponible en <http://www.albentia.com/documentacion.php> (último acceso, Agosto, 2011).

ANDRÉ, M. The sperm whale sonar: Monitoring and use in mitigation of anthropogenic noise effects in the marine environment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 602(1), páginas 262 – 267, 2009. ISSN 0168-9002. Proceedings of the 3rd International Workshop on a Very Large Volume Neutrino Telescope for the Mediterranean Sea.

ANDRÉ, M., MORELL, M., MAS, A., SOLE, M., VAN DER SCHAAR, M., ALVAREZ BAQUERIZO, C. y RODRÍGUEZ ROCH, L. *Buenas prácticas en la gestión, evaluación y control de la contaminación acústica subacuática*. Laboratorio de Aplicaciones Bioacústicas (LAB); Universidad Politécnica de Cataluña (UPC), 2009. Disponible en <http://www.lab.upc.es> (último acceso, Julio, 2011).

ANDRÉ, M. y NACHTIGALL, P. Electrophysiological measurement of hearing in marine mammals. *Aquatic Mammals. Aquatic Mammals Western Illinois University-Quad Cities*, 2007.

ANDRÉ, M., VAN DER SCHAAR, M., ZAUGG, S., HOUÉGNIGAN, L., SÁNCHEZ, A. y CASTELL, J. Listening to the deep: Live monitoring of ocean noise and cetacean acoustic signals. *Marine Pollution Bulletin*, vol. 63(1-4), páginas 18 – 26, 2011. ISSN 0025-326X. Cetaceans and military sonar: A need for better management.

ANDRÉ, M., VALLEZ, S., GERVAISE, A., C. SÁNCHEZ, VAN DER SCHAAR, M., MAS, A. y MORELL, M. Monitoring the ocean noise interactions. En *Proceedings of the 22nd Annual Conference of the European Cetacean Society, Egmond and Zee, Holanda*, página 78. European Cetacean Society, 2008.

CHAKERES, I. D. y BELDING-ROYER, E. M. AODV routing protocol implementation design. En *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04) - Volume 7*, ICDCSW '04, páginas 698–703. IEEE Computer Society, Washington, DC, USA, 2004. ISBN 0-7695-2087-1.

- CHAKERES, I. D. y KLEIN-BERNDT, L. AODVjr, AODV simplified. *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, páginas 100–101, 2002. ISSN 1559-1662.
- DAS, S., PERKINS, C. y ROYER, E. Performance comparison of two on-demand routing protocols for ad hoc networks. En *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, páginas 3 –12 vol.1. 2000.
- DOMINGO-ALADRÉN, M. C. *Diferenciación de servicios y mejora de la supervivencia en redes ad hoc conectadas a redes fijas*. Phd, Universitat Politècnica de Catalunya, 2005.
- ENGÅS, A., LØKKEBORG, S., ONA, E. y SOLDAL, A. *Effects of seismic shooting on catch and catch-availability of cod and haddock*. Fisken OG Havet No.9, 1993.
- FITZ, S., GONZALEZ-VELAZQUEZ, A., HENNING, I. y KHAN, T. Experimental investigation of wireless link layer for multi-hop oceanographic-sensor networks. En *ELECTRONICS LETTERS 24th November 2005 Vol. 41 No. 24*. 2005.
- GARROPO, R. G., GIORDANO, S. y IACONO, D. Experimental and simulation study of a WiMAX system in the sea port scenario. En *Proceedings of the 2009 IEEE international conference on Communications, ICC'09*, páginas 4729–4733. IEEE Press, Piscataway, NJ, USA, 2009. ISBN 978-1-4244-3434-3.
- GOMEZ, C., SALVATELLA, P., ALONSO, O. y PARADELLS, J. Adapting AODV for IEEE 802.15.4 mesh sensor networks: theoretical discussion and performance evaluation in a real environment. En *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks, WOWMOM '06*, páginas 159–170. IEEE Computer Society, Washington, DC, USA, 2006. ISBN 0-7695-2593-8.
- GORDON, J. y MASCROP, A. *Underwater noise pollution and its significance for whales and dolphins*. Mark Simmonds and JD Hutchinson (eds) The Conservation of Whales and Dolphins, 1996.
- GUERRA, A., GONZÁLEZ, A. y ROCHA, F. A review of record of giant squid in the north-eastern atlantic and severe injuries in architeusthis dux stranded after acoustic exploration. *ICES-Annual Science Conference, Vigo*, 2004.
- HERNANDO-RÁBANOS, J. M. *Transmisión por radio*. Editorial Universitaria Ramón Areces, 2008.
- HOUÉGNIGAN, L., ZAUGG, S., VAN DER SCHAAR, M. y ANDRÉ, M. Space-time and hybrid algorithms for the passive acoustic localisation of sperm whales and vessels. *Applied Acoustics*, vol. 71(11), páginas 1000 – 1010, 2010. ISSN 0003-682X. Proceedings of the 4th International Workshop on Detection, Classification and Localization of Marine Mammals Using Passive Acoustics and 1st International Workshop on Density Estimation of Marine Mammals Using Passive Acoustics.
- KAWADIA, V., ZHANG, Y. y GUPTA, B. System services for ad-hoc routing: architecture, implementation and experiences. En *Proceedings of the 1st international conference on Mobile systems, applications and services, MobiSys '03*, páginas 99–112. ACM, New York, NY, USA, 2003.

- KLEIN-BERNDT, L. *A quick guide to AODV routing*. Wireless Communications Technologies Group; National Institute of Standards and Technology, 2001. Disponible en [http://www.antd.nist.gov/wctg/aodv\\_kernel/](http://www.antd.nist.gov/wctg/aodv_kernel/) (último acceso, Abril, 2011).
- KLEIN-BERNDT, L. y CHAKERES, I. D. *AODVjr. A simplified version of AODV*. Mobility Management and Networking Laboratory UC Santa Barbara; National Institute of Standards and Technology, 2002. Disponible en <http://www.ianchak.com/> (último acceso, Mayo, 2011).
- KONG, P., PATHMASUNTHARAM, J., WANG, H., GE, Y., ANG, C., WEN, S., ZHOU, M. y HARADA, H. A routing protocol for WiMAX based maritime wireless mesh networks. En *VTC Spring'09*. 2009a.
- KONG, P., WANG, H., GE, Y., ANG, C., PATHMASUNTHARAM, J., WEN, S., ZHOU, M. y HARADA, H. Distributed adaptive time slot allocation for WiMAX based maritime wireless mesh networks. En *WCNC'09*, páginas 1694–1699. 2009b.
- KONG, P., WANG, H., GE, Y., ANG, C., WEN, S., PATHMASUNTHARAM, J., ZHOU, M. y DIEN, H. A performance comparison of routing protocols for maritime wireless mesh networks. En *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, páginas 2170 –2175. 2008. ISSN 1525-3511.
- MCCAULEY, R., FEWTRELL, J., DUNCAN, A., JENNER, C., JENNER, M., PENROSE, J., PRINCE, R., ADHITYA, A., MURDOCH, J. y MCCABE, K. Marine seismic surveys - a study of environmental implications. *Australian Petroleum Production and Exploration Association Journal*, vol. 40, 2000. ISSN 692-708.
- MCLARNON, B. VHF/UHF/Microwave Radio Propagation: A Primer for Digital Experimenter. *ICTP-URSI Italia*, 2001.
- PALAZZI, C., WOODS, G., ATKINSON, I. y KININMONTH, S. High speed over ocean radio link to Great Barrier Reef. En *Proceedings of IEEE TENCON '05 Conference In: Tencon 2005 - 2005 IEEE Region 10 Conference , 21 - 24 November 2005, Melbourne, VIC, Australia*. 2005. ISSN 0-7803-9312-0.
- PATHMASUNTHARAM, J., KONG, P., ZHOU, M., GE, Y., WANG, H., ANG, C., WEN, S. y HARADA, H. TRITON: High speed maritime mesh networks. En *PIMRC'08*, páginas 1–5. 2008.
- PÉREZ, E., PRIETO, D. y SOTO, I. Aplicación de la tecnología de acceso WiMAX en entornos marítimos costeros. En *XVI Jornadas TELECOM-ID 2006, Madrid*. Quobis Networks, 2006.
- PERKINS, C. y ROYER, E. Ad-hoc on-demand distance vector routing. En *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, páginas 90 –100. 1999.
- PERKINS, C. E., BELDING-ROYERS, E. y DAS, S. R. Ad-hoc on-demand distance vector (AODV) routing. *Request For Comments Experimental 3561, Internet Engineering Task Force*, 2003.
- RADHA, G. S. V., RAO, K., PURNACH, M. y RAO, R. Influence of sea breeze on line-of-sight links located in visakhapatnam. 2005.

- REMONDO, D. y NIEMEGEERS, I. G. Ad hoc networking in future wireless communications. *Computer Communications*, vol. 26(1), páginas 36 – 40, 2003. ISSN 0140-3664.
- RICHARDSON, W., GREENE JR, C., MALME, C. y THOMSON, D. *Marine mammals and noise*. Academic Press, San Diego, CA, 1995.
- ROYER, E. y TOH, C.-K. A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, vol. 6(2), páginas 46 –55, 1999. ISSN 1070-9916.
- SIEBER, A., COCCO, M., MARKERT, J., WAGNER, M. F., BEDINI, R. y DARIO, P. ZigBee based buoy network platform for environmental monitoring and preservation: Temperature profiling for better understanding of mucilage massive blooming. En *WISES*, páginas 1–14. 2008.
- SIEBER, A., WAGNER, M. F., MARKERT, J. y WOGERER, C. *Low power sensor platform for environmental monitoring*, páginas 1804–1809. IEEE SENSORS 2009 Conference, 2009.
- SIMÓ, J., OSUNA, P., SEOANE, J. y MARTÍNEZ, A. Desarrollo de nodos mesh Wi-Fi autónomos como soporte a redes de telemedicina rural en zonas aisladas. En *Proceedings of CASEIB2005. Madrid*. 2005.
- SIMÓ, J., OSUNA, P., SEOANE, J. y MARTÍNEZ, A. An IEEE 802.11 static mesh network design for isolated rural areas in developing regions. En *The 25th International Conference on Computer Communications (IEEE INFOCOM 2006). Poster Session. Barcelona - Spain*. 2006a.
- SIMÓ, J., SEOANE, J. y DE SALAZAR, R. A QoS-Aware ad hoc wireless network for isolated rural environments. En *EUNICE 2005: Networks and Applications Towards a Ubiquitously Connected World* (editado por C. Kloos, A. Marín y D. Larrabeiti), vol. 196 de *IFIP International Federation for Information Processing*, páginas 157–172. Springer Boston, 2006b.
- SKALSKI, J., PEARSON, W. y MALME, C. Effects of sounds from a geophysical survey device on catch-per-unit-effort in a hook-and-line fishery for rockfish. (*Sebastes* ssp.). *Canadian Journal of Fisheries and Aquatic Sciences*, 1992. ISSN 1357-1365.
- WANG, W., YOUN, J.-H. y SHARIF, H. The implementation of an energy balanced routing protocol with dynamic power scaling in tinyos. En *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on*, vol. 2, páginas 262 –267. 2006.
- YAPICIOGLU, T. y OKTUG, S. F. Effect of wave height to connectivity and coverage in sea surface wireless sensor networks. En *International Symposium on Computer and Information Sciences*, páginas 328–333. 2009.
- ZAUGG, S., VAN DER SCHAAAR, M., HOUÉGNIGAN, L., GERVAISE, C. y ANDRÉ, M. Real-time acoustic classification of sperm whale clicks and shipping impulses from deep-sea observatories. *Applied Acoustics*, vol. 71(11), páginas 1011 – 1019, 2010. ISSN 0003-682X. Proceedings of the 4th International Workshop on Detection, Classification and Localization of Marine Mammals Using Passive Acoustics and 1st

International Workshop on Density Estimation of Marine Mammals Using Passive Acoustics.

ZENNARO, M., BAGULA, A., GASCÓN, D. y NOVELETA, A. Planning and deploying long distance wireless sensor networks: The integration of simulation and experimentation. En *Ad-Hoc, Mobile and Wireless Networks* (editado por I. Nikolaidis y K. Wu), vol. 6288 de *Lecture Notes in Computer Science*, páginas 191–204. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-14785-2\_15.

ZHOU, M., HARADA, H., KONG, P. y PATHMASUNTHARAM, J. Interference range analysis and scheduling among three-hop neighborhood in maritime WiMAX mesh networks. En *WCNC'10*, páginas 1–6. 2010.